Spring 2021

Existence of Mutual Stabilization in Chaotic Neural Models

John Parker
University of New Hampshire, Durham

Existence of Mutual Stabilization in Chaotic Neural Models

BY

John E. Parker

BS, Elon University, 2014

DISSERTATION

Submitted to the University of New Hampshire

in Partial Fulfillment of

the Requirements for the Degree of

Doctor of Philosophy

in

Integrated Applied Mathematics

May, 2021

This dissertation has been examined and approved in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Integrated Applied Mathematics by:

Kevin Short, Dissertation Director
Professor of Mathematics
Integrated Applied Mathematics
Department of Mathematics and Statistics

John Gibson
Associate Professor of Mathematics
Integrated Applied Mathematics
Department of Mathematics and Statistics

Mark Lyon
Associate Professor of Mathematics
Integrated Applied Mathematics
Department of Mathematics and Statistics

Harish Vashisth
Associate Professor of Chemical Engineering
Integrated Applied Mathematics
Department of Chemical Engineering

Matthew Morena
Assistant Professor of Mathematics
Department of Mathematics
Christopher Newport University

On 16 April 2021

Original approval signatures are on file with the University of New Hampshire Graduate School.

DEDICATION

To Maddie, my best friend, closest ally, and partner.

ACKNOWLEDGEMENTS

I have never written an acknowledgements section before, nor have I ever written a thesis. Similar to an actor winning an Academy Award, there are many, many people who I would like to thank that helped me get to where I am now and I am sure I will forget to mention some of those people.

I would like to thank the Integrated Applied Mathematics (IAM) program within the Department of Mathematics & Statistics at the University of New Hampshire. The IAM program provided me the opportunities to learn the skills I needed to write and complete this thesis. Also, I would like to thank the IAM program for providing me financial support in the form of a teaching assistantship during my tenure at UNH. Therefore, I would like to thank the Department of Physics and Astronomy, especially Professor Dawn Meredith, for all that I learned while acting as a TA for many physics courses.

I am very grateful to my advisor, Professor Kevin Short. I would not be in the spot I am now, nor feel as prepared as a researcher without his guidance, advice, and clever insight. Additionally, I would like to thank Professor Short for his constant enthusiasm for research and teaching, curiosity for probing new ideas, and perspectives that he shared about his own experiences that was always evident during our many research meetings.

I would like to thank my committee members, Professors John Gibson, Mark Lyon, Harish Vashisth, and Matthew Morena. The various questions, feedback, and useful commentary that I received from these professors greatly impacted and improved this thesis. I would like to thank them for their added flexibility and patience during this difficult spring 2021 semester.

Finally, I would like to thank my friends and family for all of their support over these last few years. My parents were constantly supportive and only encouraged me, and I am grateful for all that they have done for me. It would be remiss if I did not mention Maddie. I am forever indebted to Maddie for her continued and constant friendship, partnership, and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Existence of Mutual Stabilization in Chaotic Neural Models

by

John E. Parker

University of New Hampshire, May, 2021

Recent work has demonstrated that interacting chaotic systems can establish persistent, periodic behavior, called mutual stabilization, when certain information is passed through interaction functions. In particular, this was first shown with two interacting cupolets (*C*haotic *U*nstable *P*eriodic *O*rbit-*lets*) of the double scroll oscillator. Cupolets are highly accurate approximations of unstable periodic orbits of a chaotic attractor that can be generated through a control scheme that repeatedly applies perturbations along Poincaré sections. The decision to perturb or not to perturb the trajectory is determined by a bit in a binary control sequence. One interaction function used in the original cupolet research was based on integrate-and-fire dynamics that are often seen in neural and laser systems and was used to demonstrate mutual stabilization between two double scroll oscillators. This result provided the motivation for this thesis where the stabilization of chaos in mathematical models of communicating neurons is investigated.

This thesis begins by introducing mathematical models of neurons and discusses the biological realism of the models. Then, we consider the two-dimensional FitzHugh-Nagumo (FHN) neural model and we show how two FHN neurons can exhibit chaotic behavior when

communication is mediated by a coupling constant, g, representative of the synaptic strength between the neurons. Through a bifurcation analysis, where the synaptic strength is the bifurcation parameter, we analyze the space of possible long-term behaviors of this model. After identifying regions of periodic and chaotic behavior, we show how a synaptic sigmoidal learning rule transitions the chaotic dynamics of the system to periodic dynamics in the presence of an external signal. After the signal passes through the synapse, synaptic learning alters the synaptic strength and the two neurons remain in a persistent, mutually stabilized periodic state even after the signal is removed. This result provides a proof-of-concept for chaotic stabilization in communicating neurons.

Next, we focus on the 3-dimensional Hindmarsh-Rose (HR) neural model that is known to exhibit chaotic behavior and bursting neural firing. Using this model, we create a control scheme using two Poincaré sections in a manner similar to the control scheme for the double scroll system. Using the control scheme we establish that it is possible to generate cupolets in the HR model. We use the HR model to create neural networks where the communication between neurons is mediated by an integrate-and-fire interaction function. With this interaction, we show how a signal can propagate down a unidirectional chain of chaotic neurons. We further show how mutual stabilization can occur if two neurons communicate through this interaction function. Lastly, we expand the investigation to more complicated networks including a feedback network and a chain of neurons that ends in a feedback loop between the two terminal neurons. Mutual stabilization is found to exist in all cases. At each stage, we comment on the potential biological implications and extensions of these results.

CHAPTER 1
INTRODUCTION

This thesis focuses on nonlinear systems of coupled, ordinary differential equations (ODEs) that define dynamical systems to describe the behavior of biological neurons. In each differential equation, the independent variable is time. The dependent variable in each ODE in the dynamical system is called a dynamical variable. Dynamical systems theory has been used to model many real-world systems including systems found in biology (e.g. predator-prey models), physics (e.g. pendulum), or epidemiology (e.g. SIR or SEIR models of epidemics) to name a few. When a dynamical system is modeling a biological, physical, or other real-world process, the dynamical variables represent some quantity of interest about the system. The systems of interest in this thesis are biological neurons, specifically interaction and signaling between neurons. Allowing dynamical models of neurons to interact, we show how novel behavior can occur and discuss the biological implications of the mathematical analysis. The rest of this chapter introduces the behavior of interacting neurons from the perspective of dynamical systems theory and the motivation for the specific questions we investigate in this thesis.

1.1 Dynamical Systems

The neural interactions of interest here are represented by coupled differential equations suitable for computational simulations. The results of the simulations are then used for careful analysis and classification of different types of behaviors in §2. When the systems are nonlinear, it is often the case that adjusting parameters produces a wide range of long-term

behaviors for the system; two particularly interesting types of behaviors for this study are periodic and chaotic dynamics.

For our numerical calculations we deem a system as periodic when the time series simulation repeats itself over a fixed number of time steps for each dynamical variable. Mathematically speaking, a vector function $\vec{u}(t)$ is periodic in t with a period L if $\vec{u}(t) = \vec{u}(t+nL)$ for any non-zero positive integer n for all time t.

Chaotic dynamics occur when the dynamical variables never settle into a periodic pattern, yet remain bounded for all time. Several further properties separate chaotic behavior from quasi-periodicity (or almost periodic behavior). Hunt & Ott [1] examine and compare different definitions of chaos, provide several examples of chaotic systems, and proposes a new definition for chaos. Chaos is often characterized by the rate of divergence for two initially close, but separated trajectories. This rate of divergence is related to the Lyapunov exponents of the system, and a system can be generally classified by the maximum Lyapunov exponent (MLE) of the system. For chaotic behavior the MLE is positive; while periodic or quasi-periodic behaviors have nonpositive MLE. Additionally, when examining the system in phase space (as described in §2) the difference between chaotic and periodic behavior is easily distinguished by the viewer.

1.2 Motivations

Understanding why and how dynamical systems can produce chaotic behavior, yet remain in a bounded domain has been an area of study since Poincaré first demonstrated this behavior with the three body problem [2]. Recent work in chaotic dynamical systems has shown how the control of chaotic dynamics can result in novel behavior and insight into the chaotic system. In Parker [3], the authors discuss a control method on a chaotic system called the double scroll oscillator. The control method works to stabilize the unstable periodic orbits typically traversed by the dynamics of a chaotic attractor. The stabilized orbits are called cupolets (*C*haotic *U*nstable *P*eriodic *O*rbit-*LETS*) and would be unstable without the

control scheme [3–6]. Different controls produce different cupolets, and for almost all cases there is a unique map between the control sequence and the final cupolet that is produced. Further, the resulting cupolet will be produced no matter what initial condition is used. The implications of the work in Morena & Short [7] is that two systems that would naturally be in disordered chaotic states can mutually stabilize into linked, ordered periodic behavior.

Additionally, it has been shown how two chaotic systems can transmit information about one system to the other through an interaction function. In particular, one interaction function of interest was borrowed from neural dynamics, the *integrate-and-fire* function. In this mutual stabilization the chaotic systems leave the chaotic regime and stabilize onto a linked pair of periodic cupolets. The method is described in detail in Morena & Short [7] and Short & Morena [8], and it provided the motivation to see if a similar stabilization could occur in a system of chaotic neurons. We describe the stabilization process from Short & Morena [8] below in Figure 1.1.

In Figure 1.1a two dynamical systems evolve in a chaotic regime. A control sequence is applied to Chaotic System 1, shown as the double scroll oscillator. Controls are applied only along the control planes (seen edge on) marked 0 and 1, and represent small kicks along the line of intersection. The kicks follow a prescribed small perturbation that depends on the position along the line of intersection. The controls are encoded into a sequence of 0's and 1's where a 0 represents essentially no control applied and a 1 represents a large control or a macrocontrol. The "kick" or "no kick" control sequence is applied as the trajectory of the double scroll oscillator intersects with the designated control plane. These control planes are labeled as 0 and 1 in Figure 1.1a. This control scheme, adapted from Hayes *et al.* [9] and Hayes *et al.* [10], is described in more detail in Short & Morena [8]. The control sequence is applied and generates a cupolet on System 1, and we show one such stabilized cupolet, Cupolet 1, in Figure 1.1b. Cupolet 1 then visits the 0 and 1 control planes periodically, and this sequence is then passed into the interaction function (e.g., an *integrate-and-fire* function) that generates a control sequence to interact with Chaotic System 2. The controls

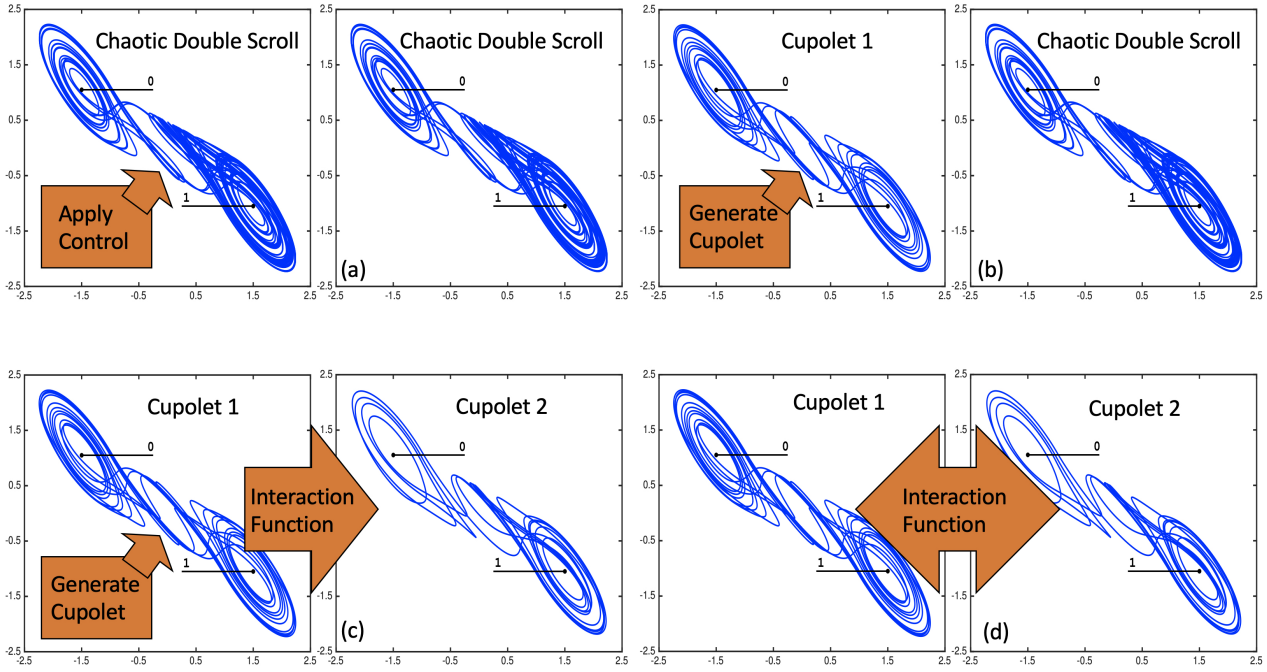

Figure 1.1: Diagrams outlining the process of generating mutual stabilization between interacting chaotic systems through cupolet communication. This example is based on Short & Morena [8]. (a) A control sequence is applied to Chaotic System 1 and the double scroll oscillator with Chaotic System 2 evolves naturally. (b) A cupolet, Cupolet 1, is generated from the control sequence on Chaotic System 1 while Chaotic System 2 continues to evolve. (c) Cupolet 1 sends its trajectory information through the interaction function as a control onto Chaotic system 2, which generates a cupolet, Cupolet 2. (d) Cupolet 1 passes its trajectory information through the interaction function as a control onto Cupolet 2, allowing Cupolet 2 to persist. Cupolet 2 passes its trajectory information through the interaction function as a control on Cupolet 1, allowing Cupolet 1 to persist. This replaces the original control in (a). The two systems are now mutually stabilized with each other.

coming out of the interaction function then act on Chaotic System 2, and thus a second cupolet, Cupolet 2, is stabilized, as shown in Figure 1.1c. At this point the feedback part of the process occurs, and Cupolet 2 periodically visits the 0 and 1 control planes, generating a sequence that is passed into the interaction function to generate a control sequence that is fed back to Cupolet 1. In numerous cases, the control sequence feeding back from Cupolet 2 is exactly that which is needed to stabilize Cupolet 1, and so this control replaces the original external control on System 1 and the net result is that Cupolet 1 and Cupolet 2 become mutually stabilizing. In Figure 1.1d both cupolets are sending control sequences through the interaction function and the cupolets shown are one of the many pairs of mutually stabilized, phase locked, persistent periodic states that can occur for the double scroll oscillator.

In Morena & Short [7], the authors show how the above mutual stabilization can occur with different mutually stabilized cupolets as well as with several different types of interaction functions, including *integrate-and-fire* dynamics based on neural firing. This result gives the motivation for the question asked in this dissertation: "Can chaotic neural systems lead to mutual stabilization through properly constructed interaction functions?" In order to investigate this further, we must first establish a neural dynamical system capable of producing chaos, along with suitable interaction functions for these systems.

1.3 Neural Models

Various dynamical systems exist to represent the electrochemical signal produced by a neuron. One of the most well known and biologically relevant models is called the Hodgkin-Huxley model. The Hodgkin-Huxley (HH) model, originally published in 1952 [11], is a four dimensional dynamical system modeling the neural firing, or action potential, of a neuron. In the HH model, each parameter represents a physical or chemical process known to be associated with cellular processes. The Hodgkin-Huxley dynamical system as original stated in Hodgkin & Huxley [11] is given in Eq 1.1.

$$
\begin{aligned}
I &= C_M \frac{dV}{dt} + \bar{g}_K n^4 (V - V_K) + \bar{g}_{Na} m^3 h (V - V_{Na}) + \bar{g}_l (V - V_l) \\
dn/dt &= \alpha_n (1 - n) - \beta_n n \\
dm/dt &= \alpha_m (1 - m) - \beta_m m \\
dh/dt &= \alpha_h (1 - h) - \beta_h h
\end{aligned}
\tag{1.1}
$$

The variable I is a constant current that is applied to the membrane potential. The different subscripts correspond to the respective ion channels commonly seen in the membrane of a neuron. The "K" subscript refers to the current from the potassium ions, the "Na" subscript refers to the current from the sodium ions, and the "l" subscript refers to the leakage current. The "$\bar{g}$" terms are the conductance of the respective current channel. The main dynamical variable is V which reflects the membrane potential of the neuron. The "V_l" reflect the reversal potential of the leakage current. The "V_{Na} and "V_K" reflect the reversal potentials of the sodium ion channel and potassium ion channel, respectively. The other three dynamical variables are dimensionless and reflect proportions of molecules associated with each ion channel. Each of these dimensionless variables contain an α and β parameter, that are functions of the membrane potential, considered as auxiliary functions. The functional forms are given in Eq 1.2 as originally stated in Hodgkin & Huxley [11]. Extensions of the HH model exists where more complex models reflect "Hodgkin-Huxley formalism" where more ion channels are considered as well as the respective governing auxiliary functions:

$$
\begin{aligned}
\alpha_n &= 0.01(V + 10)/(\exp(\frac{V + 10}{10}) - 1) \\
\beta_n &= 0.125 \exp(V/80) \\
\alpha_m &= 0.1(V + 25)/(\exp(\frac{V + 25}{10}) - 1) \\
\beta_m &= 4 \exp(V/18) \\
\alpha_h &= 0.07 \exp(V/20) \\
\beta_h &= 1/(\exp(\frac{V + 30}{10}) - 1).
\end{aligned}
\tag{1.2}
$$

In a previous study of a model based on the HH neural equations, it was shown that transitions between chaotic and periodic behavior can occur as a parameter is varied [12]. Several researchers have developed simpler, lower dimensional versions of the HH equations that are simpler to analyze, but preserve important spiking and refractory behaviors that characterize neuronal dynamics. This dissertation will focus on these simplified models. In 1961 and 1962, FitzHugh [13] and Nagumo *et al.* [14], respectively, developed a reduced HH model, now called the FitzHugh-Nagumo (FHN) model. The FHN model reduces the system to two dynamical variables. The two dynamical variables represent the cellular membrane potential and the recovery dynamics of the HH model. Even though the direct biological interpretation of the parameters of the HH model is not present in the FHN model, it still exhibits most of the neural firing properties of the HH model and therefore captures firing dynamics of biological neurons, making it well suited for numerical simulations. The FHN model is further examined in Chapter 2. A second model, the Hindmarsh-Rose model, will be addressed in Chapter 3.

A single FHN neuron cannot be chaotic, unless it is driven chaos as in Vaidyanathan [15] and Lai *et al.* [16], since it is known that chaos cannot occur in two dynamical dimensions for continuous systems. However, when coupled with another neuron, the system now becomes 4-dimensional and chaos can occur as shown in Campbell & Waite [17]. Taking a similar approach in §2.3, we show how the 4-dimensional system can exhibit both chaotic and periodic behavior when the parameters of the system are varied. Then, following the discussion from §1.2, we show how mutual stabilization can arise within the system through neural learning so that a system, originally in a chaotic state, can stabilize into linked periodic behavior. We start with a simple model in Chapter 2 and show the different behavioral modes that can be attained by varying the coupling strength between the neurons. Then, by treating one of the parameters as a dynamical variable to allow for synaptic learning and by allowing that parameter to evolve, we will show how it is possible to push the model toward any desired behavior, whether periodic or chaotic, in Chapter 2.

The sections §1.1-§1.3 of this chapter has been adapted and reproduced from Parker, J. E. & Short, K. M. Sigmoidal synaptic learning produces mutual stabilization in chaotic FitzHugh-Nagumo model. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **30,** 063108 (June 2020), with the permission of AIP Publishing.

CHAPTER 2
THE FITZHUGH-NAGUMO MODEL

2.1 Physical Model

The FHN system we will study in this chapter is a bidirectional, two-cell model. The physical analog assumes a long chain of neurons that terminates at the two coupled neurons in our model. A simple diagram is given in Figure 2.1, which shows how Neuron 1 (N1) receives an external signal from a chain of neurons in addition to feedback from Neuron 2 (N2). This signal propagates down the chain until it enters the cell body of N1. The signal then integrates with the membrane potential until the firing threshold is reached, at which point the neuron fires a signal down the axon. When the signal reaches N2 it begins to integrate into N2's cell body, simply repeating the same process of building to a firing threshold before releasing a signal down the N2 axon feeding back to N1. We note that in our model N1 has two external inputs - the signal propagating from the chain of neurons and the feedback from N2, whereas N2 only has an external input of N1's outgoing signal.

2.2 Mathematical Model

The original FHN model of FitzHugh [13] and Nagumo *et al.* [14] can exhibit sustained oscillations when a constant direct current I_0 is applied. The version of the FHN model in this chapter is an outgrowth of the scaled FHN model in Brown *et al.* [19], that was further adapted in Rehan & Hong [20]. Additional adaptations of this model have been seen in Lai *et al.* [16] and Vaidyanathan [15]. The adapted model in Eq 2.1 can also show sustained oscillations, like the original FHN model, but with an oscillatory external electrical

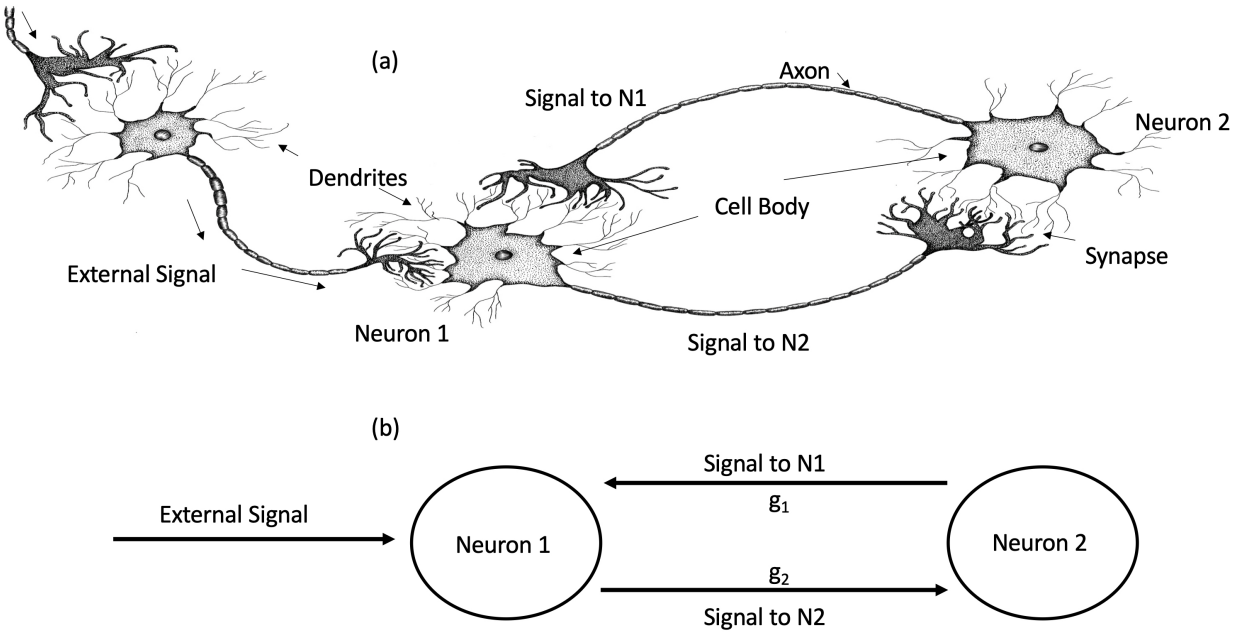

Figure 2.1: Diagram depicting the interaction of neurons. Neuron 1 (N1) receives an external signal from a chain of neurons. N1 fires an action potential to N2. N2 is coupled back with N1 via its own axon. (a) Artist's biological interpretation of the coupled system. (b) Simplified diagram reflecting mathematical coupling.

stimulation (EES) of the form $I_0(t) = A/\omega \cos(\omega t)$. The model used in Eq 2.1 follows from Rehan & Hong [20] where two FHN neurons were coupled under EES. We consider the case where only one FHN neuron is under EES and the recovery variable is dependent on one dynamical variable. Figure 2.2 shows a numerical example of the v time series (Eq 2.1) when an EES of $I_0(t) = A/\omega \cos(\omega t)$ is applied.

In Eq 2.1, the dynamical variable v represents the membrane potential of the neuron whereas w represents the recovery dynamics of the membrane potential. The recovery dynamics act as a way to recharge the neuron in order for it to fire again. The parameter b governs the strength of the recovery variable, and is inversely related to the amplitude of v for a given fixed parameter a. The parameter a is a weighting parameter for the membrane potential, v, that vertically shifts the oscillations of the neural firing. As a increases, for a

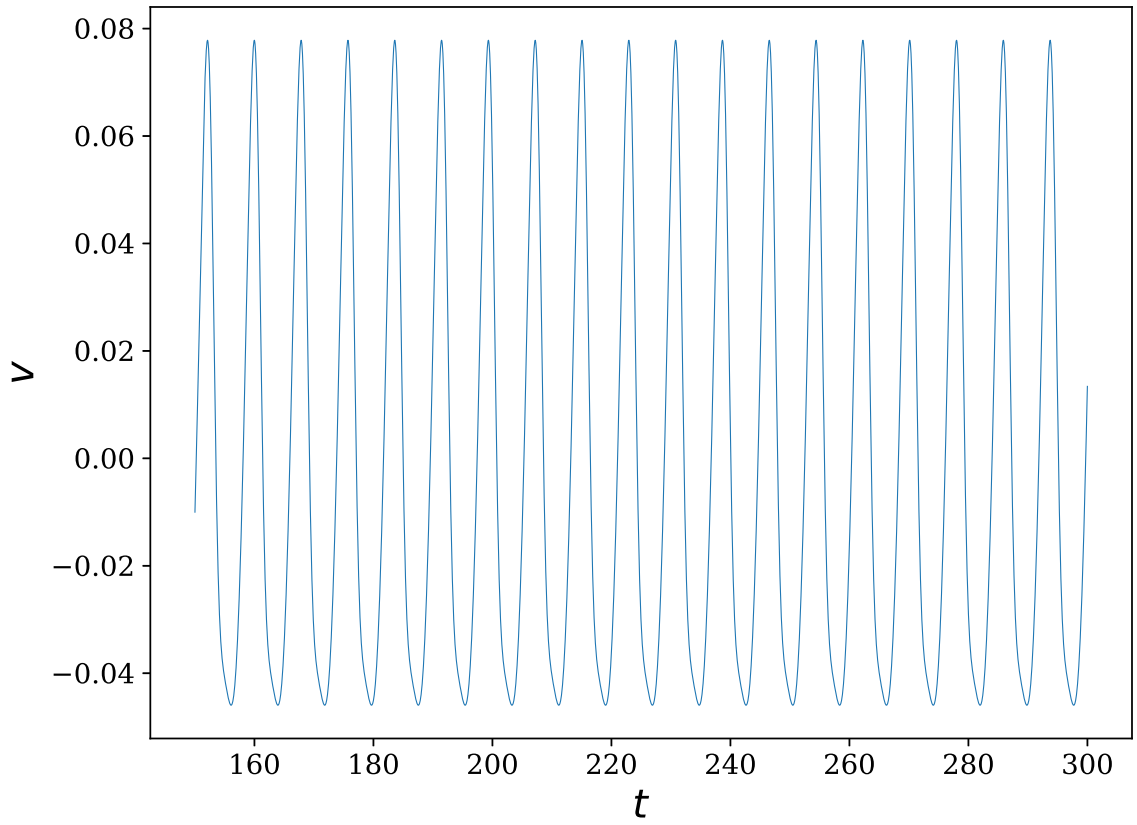

Figure 2.2: Numerical simulation of Eq 2.1 with $a = 8$ and $b = 2.5$ with an external stimulus, $I_0(t) = (A/\omega)\cos(\omega t)$ applied to $\dot{v}$ with A and ω values as given in §2.6 Figure 2.6.

fixed parameter b, the minima between oscillations of v increase.

$$\begin{aligned} \dot{v} &= v(v-1)(1-av) - w \\ \dot{w} &= bv \end{aligned} \tag{2.1}$$

In our system we will consider two of these cells, bidirectionally and electrically coupled. The implementation of the electrical coupling follows from Campbell & Waite [17] for the FHN model, and is adapted from Skinner *et al.* [21] electrical coupling of HH neurons. For similar examples of electrical coupling in FHN neurons, see Rehan & Hong [20], Chillemi *et al.* [22], Hoff *et al.* [23], and Davison *et al.* [24]. For other types of neural models with similar

electrical coupling, see Ao *et al.* [25] for HH neurons and Shuai & Durand [26] for Hindmarsh-Rose neurons. In keeping with these approaches, the electrically coupled bidirectional system studied in this chapter is given in Eq 2.2.

$$\begin{aligned} \dot{v_1} &= v_1(v_1 - 1)(1 - av_1) - w_1 + g(v_1 - v_2) \\ \dot{w_1} &= bv_1 \\ \dot{v_2} &= v_2(v_2 - 1)(1 - av_2) - w_2 + g(v_2 - v_1) \\ \dot{w_2} &= bv_2 \end{aligned} \tag{2.2}$$

The subscript on the variables corresponds to the neuron label (e.g., v_1 represents the membrane potential of N1). Here parameters a and b govern the same properties as in Eq 2.1. The coupling parameter g represents the synaptic strength of the connection between the neurons and it is multiplied by the difference between the membrane potentials. Note that when $g = 0$ the two neurons reduce to two decoupled single neurons following Eq 2.1. The system in Eq 2.2 uses the same a, b, and g values in each neuron, meaning that N1 and N2 are identical neurons. Eq 2.2 represents Figure 2.1 in the absence of external signal felt by N1. An example of Eq 2.2 dynamics are shown Figure 2.3 where the left plot shows the phase space of neuron N1 ($v_1(t)$ versus $w_1(t)$) and the right plot is the phase space of neuron N2 ($v_2(t)$ versus $w_2(t)$).

Now that the two neurons are appropriately coupled to each other, the external stimulus propagating from a chain of neurons needs to be introduced. We will call this external signal $I_0(t)$. This signal is the cumulative effect of a stimulus propagating through many neurons before reaching the N1 - N2 system. Following FitzHugh [13] and Nagumo *et al.* [14], we incorporate the external signal $I_0(t)$ as an input to the membrane potential of N1 and therefore integrate the signal into this neuron's membrane potential equation for $\dot{v_1}$. In this way the external signal directly affects the membrane potential of N1. The functional form of $I_0(t)$ may be any sort of oscillatory or pulse signal including a single action potential, chaotic, or periodic signals. In this investigation, we will focus on an oscillatory input, $I_0(t)$,

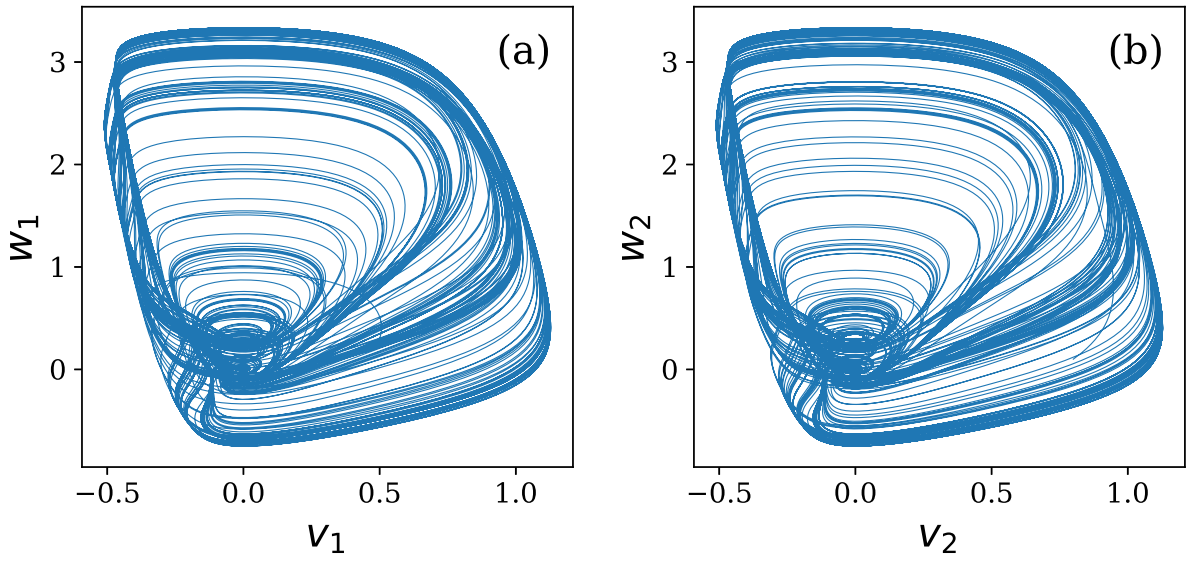

Figure 2.3: Neural dynamics of Eq 2.2 in a chaotic regime $(a, b, g) = (8, 2.5, 0.96)$ for a simulation time of $t = 0$ to $t = 2000$ with $dt = 0.01$. Both figures were numerically simulated using the numerical scheme described in §2.6. (a) Phase space of neuron N1, v_1 on the horizontal axis and w_1 on the vertical axis. (b) Phase space of neuron N2, v_2 on the horizontal axis and w_2 on the vertical axis.

modeled after the external signal in Lai *et al.* [16]. Integrating this signal into Eq 2.2, the system of equations becomes:

$$\begin{aligned}
\dot{v_1} &= v_1(v_1 - 1)(1 - av_1) - w_1 + g(v_1 - v_2) + I_0(t) \\
\dot{w_1} &= bv_1 \\
\dot{v_2} &= v_2(v_2 - 1)(1 - av_2) - w_2 + g(v_2 - v_1) \\
\dot{w_2} &= bv_2 \\
I_0(t) &= H[t - t_s] \exp(-l(t - t_s)(A/\omega) \cos(\omega(t - t_s))
\end{aligned} \tag{2.3}$$

In Lai *et al.* [16], the external signal is $I_0(t) = (A/\omega) \cos \omega t$; however, we want the signal to be felt by N1 at time $t = t_s$ and we want the signal to decay away so that we can examine the state of the dynamics of the coupled neurons before, during, and after $I_0(t)$ passes. Representing the Heaviside step function as H in Eq 2.3, the value t_s indicates when the Heaviside

function jumps from zero to one, thus allowing the system to be modeled as $I_0(t)$ turning on at $t = t_s$. We consider the case where $I_0(t)$ is not constantly active, but becomes smaller over time so we have chosen $I_0(t)$ to exhibit exponential damping, where l is the damping rate.

2.3 Parameter Search

For the cupolet behavior discussed in §1.2, two chaotic systems communicate through an interaction function. In the FHN system the synaptic strength of the neurons, g, represents this interaction function. Since dynamical systems have the capability to exhibit a variety of different behaviors as various parameter values change, we first investigate a range of synaptic strengths, g, to find what values provide desired behaviors.

We can expand on this initial analysis with a careful bifurcation diagram to investigate how the values of g produce different regions of behavior. A bifurcation diagram is a useful tool for qualitatively inspecting the changing behaviors of a system while varying one parameter (g in our model). As the parameter varies, we look at the peaks of the long-term behavior of the time series representation of the system. If the system is chaotic, there will be many peaks for the times series since the dynamics are aperiodic and will not exactly repeat for all time. However, if the system is periodic, there will only be a finite number of peaks, creating a distinctive pattern in the long term behavior. More specifically, if the system is periodic for the parameter value $g = g_0$ with N local maxima in a single period, then in the bifurcation diagram we find g_0 on the horizontal axis, and above g_0 we plot N dots at the values of each local maxima along the vertical axis. The bifurcation diagram for Eq 2.2 is given in Figure 2.4. Here, the g value is fixed for both neurons and no external signal is present. The bifurcation diagram was generated with parameters $a = 8$, $b = 2.5$ and $g \in [0.48, 1.0]$ with g varying discretely in increments of $\Delta g = 0.001$.

From visual inspection, it is easy to see regions of both periodic and chaotic behavior in Figure 2.4. For example, at $g = 0.82$ there are 5 dots plotted corresponding to periodic

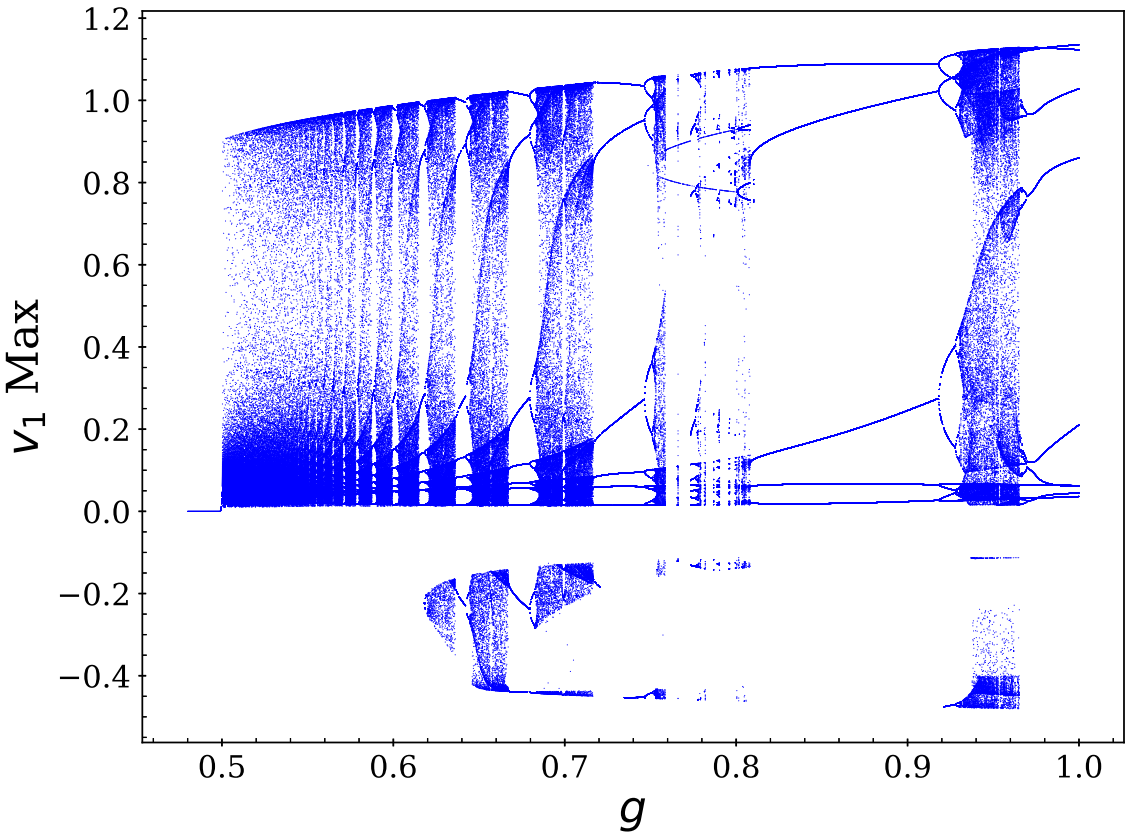

Figure 2.4: Bifurcation diagram of v_1 in Eq 2.3 with $a = 8$ and $b = 2.5$. The synaptic strength g was varied identically for both neurons. Values of $g \leq 0.5$ produced silent neurons, i.e. $v_i = 0$ for the membrane potential. Other dynamical variables produced consistent bifurcation diagrams.

behavior with 5 different maxima in each period. At $g = 0.96$ we have a region of apparent chaos. For synaptic strength $g < 0.5$, the neurons stop firing so the bifurcation diagram reflects silent long term behavior for values below 0.5. Using the bifurcation diagram, it is clear that at different fixed values of the parameter g, the interaction and feedback between the two neurons can result in chaotic behavior or it can settle into a mutually synchronized periodic behavior. The presence of chaotic and periodic behaviors for nearby values of g is an indication that synaptic learning may allow the system to evolve from a chaotic state to a periodic state, as will be discussed in §2.5. This is similar to the mutual stabilization

architecture discussed in §1.2.

In the previous study of cupolets, it was found that an *integrate-and-fire* interaction function could lead to mutual stabilization that would drive two chaotic systems into periodic behavior. To extend this to the FHN model would imply that the dynamics of the synaptic strength, g, could lead to mutual stabilization. This can be done by treating g as a dynamical variable instead of a parameter. We examine whether it is possible to treat g as an interaction function so that the two-cell FHN system will transition from chaotic to periodic behavior. Figure 2.5 shows an example where we identify two desired states that we will target in our analysis.

Initially, the system begins in a chaotic state corresponding to the parameters $(a, b, g) = (8, 2.5, 0.96)$. We target a final state that becomes periodic when $g = 0.82$. In the next section, we discuss the dynamics of the synaptic strength that leads to this desired change of state and then develop a methodology that will allow for a transition between any two states.

2.4 Synaptic Strength Dynamics

As discussed, we need the synaptic strength to evolve dynamically in order to demonstrate evidence of neural learning in the system. One of the most common types of neural learning is Hebbian learning, which is a type of neural learning often found in biological neural networks. With Hebbian learning dynamics, the synaptic strength between two neurons increases if they fire together. This can be paraphrased from Hebb's rule as *if the presynaptic cell fires and contributes to the firing of the postsynaptic cell, then the synaptic strength between the two should increase* [27]. We develop a method that stems from Hebbian learning concepts. In our model simulations this would mean that if N1 fires, and then N2 fires as a result of N1, then the synaptic strength connecting N1 to N2 should increase. Mathematically, the basic Hebb's rule in our model would be $\dot{g} = v_1 v_2$ according to Hebb [27], or that g changes as the product of the outputs of the connected neurons, N1 and N2. An extension of Hebbian

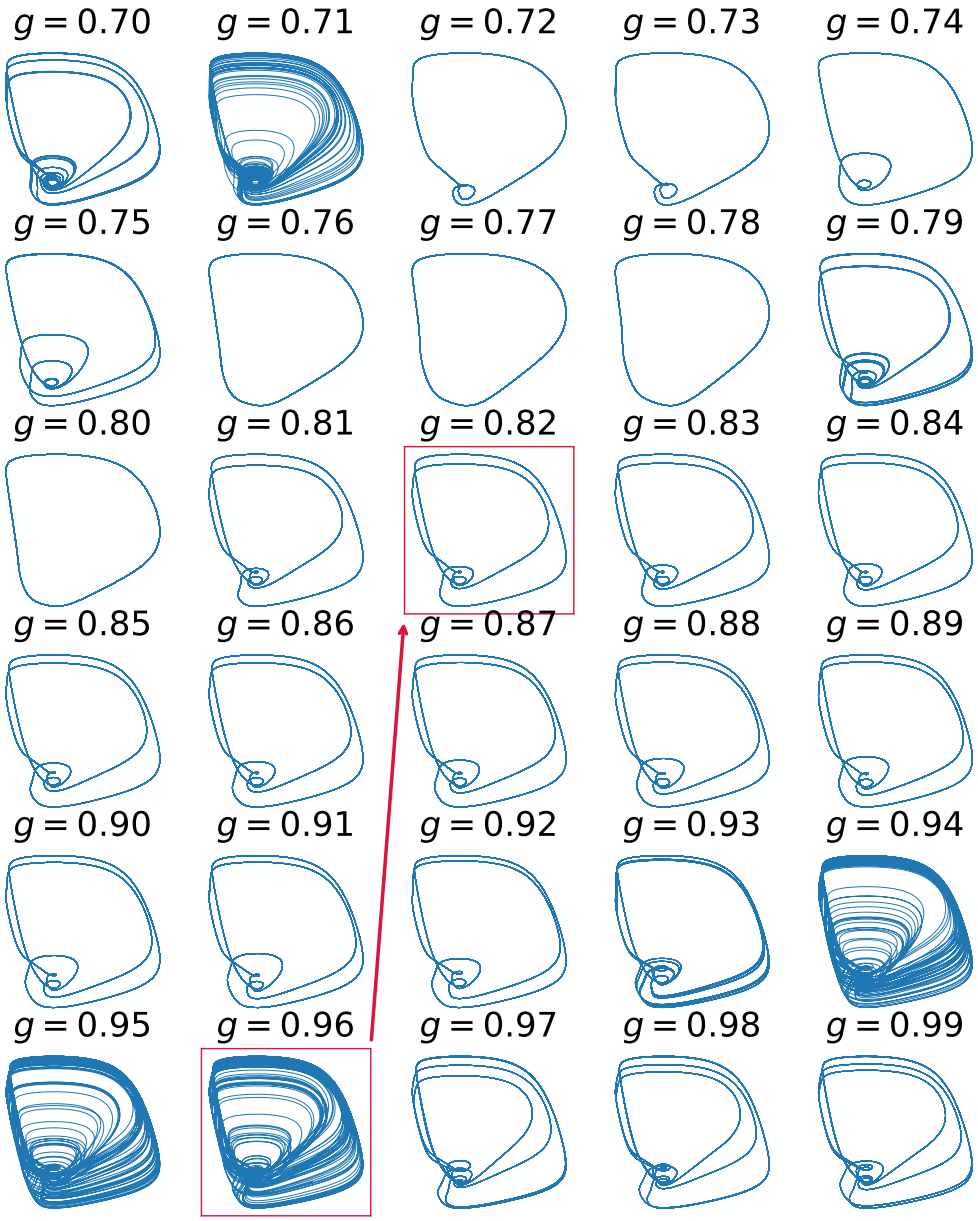

Figure 2.5: The phase plane behavior of N1 for different values of the synaptic strength, g, where each plot shows v_1 on the horizontal axis and w_1 on the vertical axis. The g values range from 0.7 to 0.99. We select $g = 0.96$ (tail of arrow) in a chaotic regime as our initial synaptic strength and we want to transition to $g = 0.82$ (tip of arrow) in a periodic regime as our desired target state through synaptic learning.

learning is known as Oja's rule, and this can be viewed as a correction term to improve Hebb's rule. If we consider the feedback communication between two neurons, from neuron N2 to neuron N1, then for this feedback channel, N2 is the presynaptic cell and N1 is the postsynaptic cell. Therefore, with Oja's rule implemented, the equations for the synaptic strength between the postsynaptic cell and the presynaptic cell are given in Eq 2.4. There are two equations since the two neurons are coupled. The first term of each equation in Eq 2.4 reflects Hebb's rule and the second term is Oja's modification of Hebb's rule [28].

$$\begin{aligned} \dot{g_1} &= \alpha(v_1 v_2 - v_1^2 g_1) \\ \dot{g_2} &= \alpha(v_2 v_1 - v_2^2 g_2) \end{aligned} \tag{2.4}$$

In Eq 2.4, v_1 represents the output of N1 and v_2 represents the output of $N2$. Therefore, the synaptic strength g would change according to these dynamics. The variable α is a small positive value corresponding to the learning rate [28]. One issue with Hebbian learning is an unbounded increase in the magnitude of the synaptic strength. That is, if each neuron's firing variables (v_1 for N1 and v_2 for N2) are the same sign then the synaptic strength will simply continue to become larger until one of the firing variables become 0. Similarly, if the dynamical variables v_1 and v_2 have the opposite sign then the synaptic strength will continue to grow smaller until one of the firing variables becomes 0. In the form of Oja's rule, Eq 2.4, the term $v_1 v_2$ will be either positive or negative depending on the values of v_1 and v_2. In our model, when the neuron fires an action potential we see a narrow spike above the horizontal axis, but otherwise the neuron oscillates around or below the horizontal axis. Therefore, since the neurons may be chaotic, the action potentials of the neurons are not synchronized and v_1 and v_2 would not generally have the same sign. Even when the neurons are traversing through regions of periodic behavior, if they are not synchronized, then v_1 and v_2 still may not have the same sign. In our simulations, this causes $\dot{g} \leq 0$ to dominate, leading to an overall decrease in the value of the synaptic strength, g. As we have seen with the bifurcation diagram in Figure 2.4, if g drops to below $g = 0.5$, the neurons will become silent and stop

firing. Therefore, if synaptic learning can be treated dynamically and controlled so that the g parameter makes the appropriate transition, then it would be possible to have the coupled neuron system transition from chaotic to periodic behavior. This will be addressed in the next section.

2.5 Sigmoidal Synaptic Learning

In order to avoid the unbounded increase in magnitude of the synaptic strength that would occur for Hebbian learning and Oja's rule, we develop a function that provides a smooth transition from one state to another and saturates at the extremes of its range. The transition is instigated by the external signal which is represented as $I_0(t)$ in Eq 2.3. Sigmoidal curves provide this desired transition where the function begins at one value and then steeply changes to another value. The logistic sigmoidal curve, $S(x)$, has the functional form of $S(x) = \frac{1}{1+\exp(-x)}$ and $S(x) \in (0,1)$ for $x \in (-\infty, \infty)$. The upper bound of the sigmoidal function can be considered to be the initial synaptic strength of the system. The sigmoidal function will be incorporated into our model so that when a signal arrives at N1, the synaptic strength will be at the upper saturation level. When N1 receives the external signal, $I_0(t)$, the presence of the signal will cause "learning" in the synapse so that after a sufficiently strong signal passes through, the synaptic strength will have reached the lower saturation level for the sigmoid.

Let us define $g(t)$ so that it is similar to $S(x) = \frac{1}{1+\exp(-x)}$, but with more control over the range of values $g(t)$ traverses. Replacing the 1 in the numerator with the variable c and the 1 in the denominator with the variable d, and the exponential function with our external function, $I_0(t)$, we then have $g(t) = \frac{c}{d+I_0(t)}$, which is not a true sigmoidal function, but has damped oscillations that still produce the desired sigmoidal envelope. However, since we want the sigmoidal envelope to begin after the external signal arrives we must impose constraints on g, given in Eq 2.5, that enforce this behavior. This allows us to consider the dynamics of g in Eq 2.6, with initial condition g_i chosen as our desired initial state from

Figure 2.5. In this implementation, we have control over the values that g can attain through choice of c and d. Eq 2.5 shows the piecewise form of $g(t)$ and Eq 2.6 shows the dynamical differential equation for the now sigmoidal $g(t)$.

$$\begin{aligned} g(t < t_s) &= g_i \\ g(t \geq t_s) &= \frac{c}{d + I_0(t)} \end{aligned} \tag{2.5}$$

$$\dot{g} = \frac{-g^2}{c} \frac{\mathrm{d}I_0(t)}{\mathrm{d}t} \text{ with } g(0) = g_i \tag{2.6}$$

Since $I_0(t)$ is exponentially damped and only nonzero after $t = t_s$, we can define the beginning and ending values of $g(t)$ based on this property of $I_0(t)$. The initial g, called g_i, will hold when $t \leq t_s$, since $\dot{g} = 0$ if $I_0(t) = 0$ as shown in Eq 2.6. Since $I_0(t)$ contains exponential damping of the form $\exp(-l(t - t_s))$, when $t \to \infty$ then $I_0(t \to \infty) \to 0$ meaning that the long-term targeted synaptic strength, g_f, is achieved as $t \to \infty$, that is $g(t \to \infty) \to g_f = \frac{c}{d}$. For simplicity, let $d = 1$ and $c = g_f$. Therefore, in order to enforce that g takes the appropriate value at $t = t_s$ we must set $I_0(t_s) = (A/\omega)$ to a specific value as shown in Eq 2.7.

$$\begin{aligned} g(t_s) = g_i &= \frac{g_f}{1 + (A/\omega)} \\ (A/\omega) &= \frac{g_f}{g_i} - 1 \end{aligned} \tag{2.7}$$

At any initial state, g_i, there are countless pairs for A and ω in order for g to trend towards the desired g_f. This implies that for any one state change, i.e. $g_i \to g_f$, there are many $I_0(t)$ functions varying in frequency and amplitude that will induce the desired state change. Since $I_0(t)$ is an exponentially damped sinusoidal function and $g(t)$ is dependent on $I_0(t)$, then $g(t)$ will have damped oscillations that asymptotically approach g_f, representing the

sigmoidal nature of $g(t)$. These adaptations of Eq 2.3 appear in Eq 2.8.

$$\begin{aligned}
\dot{v_1} &= v_1(v_1-1)(1-av_1) - w_1 + g_1(v_1 - v_2) + I_0(t) \\
\dot{w_1} &= bv_1 \\
\dot{v_2} &= v_2(v_2-1)(1-av_2) - w_2 + g_2(v_2 - v_1) \\
\dot{w_2} &= bv_2 \\
\dot{g_1} &= \frac{-g_1^2}{g_f}\frac{\mathrm{d}I_0(t)}{\mathrm{d}t} \\
\dot{g_2} &= \frac{-g_2^2}{g_f}\frac{\mathrm{d}I_0(t)}{\mathrm{d}t} \\
I_0(t) &= H[t-t_s](A/\omega)\cos(\omega(t-t_s))\exp(-l(t-t_s))
\end{aligned} \tag{2.8}$$

With this implementation of synaptic learning we have the external signal, $I_0(t)$, as the driver of the system from one state to the next. If $I_0(t)$ remained 0 in Eq 2.8, the state maintains the initial state with no g variation. Additionally, the external signal is modulated by the frequency ω and the amplitude A. From the dynamical equations in Eq 2.8, it can be seen that as long as the ratio (A/ω) is the same, the synaptic strength will always be driven to the same g_f value. Further, the method is general so that any initial and final state can be selected.

2.6 Results

Numerical simulation of Eq 2.8 leads to three types of states within one set of parameters. These three states are *Initial*, *Transient*, and *Final*. Each state is defined by a change in $I_0(t)$ propagating to N1. First, define the *Initial* state as the state of the system before $I_0(t)$ propagates to N1. The *Transient* state occurs when $I_0(t)$ first becomes nonzero and continues while it is above a designated minimal threshold. The *Final* state begins at the end of the *Transient* state and continues until the simulation concludes. To simulate the system, Eq 2.8 was numerically evaluated for $0 \leq t \leq 12000$ where $t_s = 2000$. The value

of t_s was chosen such that the system could fully evolve in the chaotic regime prior to the onset of $I_0(t)$. The simulation was performed using Python's (Version 2.7.15, Python Software Foundation) SciPy and NumPy libraries [29–31]. We used SciPy's odeint command to numerically integrate the system with a time step of $dt = 0.01$. Smaller values, (e.g. $dt = 0.001$) gave equivalent results. SciPy's odeint command uses the LSODA ordinary differential equation solver to numerically integrate the system [29, 32, 33]. The *Initial* state then occurs when $0 \leq t \leq 2000$. The time period from $I_0(t)$ activation at $t_s = 2000$ until the exponential damping term $\exp(-l(t - t_s)) = 0.0005$, is designated as the *Transient* state to imply when $I_0(t)$ is non-negligible. We note that with a stronger damping rate, l, the same dynamics would be seen but the *Transient* state would be shorter in duration. After this point $I_0(t)$ is negligible and no longer impacts the system in any appreciable way. In the simulations $I_0(t)$ always had the same decay rate and therefore the *Transient* states occurred for $2000 < t \leq 9600.91$. The remaining simulation time is designated as the *Final* state, $t > 9600.91$ until the simulation ends at $t = 12000$.

To demonstrate the dynamics of the coupled model in Eq 2.8, we provide the simulation results for the case where $g_i = 0.96$ and $g_f = 0.82$. In this case the *Initial* state ($g_i = 0.96$) is chaotic and the *Final* state ($g_f = 0.82$) has transitioned to periodic behavior. Figure 2.6 shows a three dimensional plot comparing how the *Initial* state is in the chaotic regime and the *Final* state is periodic. Similarly, Figure 2.7 shows the *Initial*, *Transient*, and *Final* state of the two coupled neurons with N1 in the left column and N2 in the right column. The rows represent the *Initial* (top row), *Transient* (middle row), and *Final* (bottom row) states. The initial conditions were the same as in Figure 2.6 with values $(v_1, w_1, g_1, v_2, w_2, g_2) = (0.5, 0.3, 0.96, 0.8, 0.1, 0.96)$, as were the parameters $(a, b, t_s, A, \omega) = (8, 2.5, 2000, 0.0217, 2\pi f)$ with $f = 0.1271$. It is clear in Figure 2.6 and Figure 2.7 that the *Initial* state is chaotic whereas the *Final* state has settled into periodic behavior. The *Final* states of the neurons are identical in phase space. The time series v_1 and v_2 both have a period of 27.28 seconds, or 2728 samples, and are out of phase with

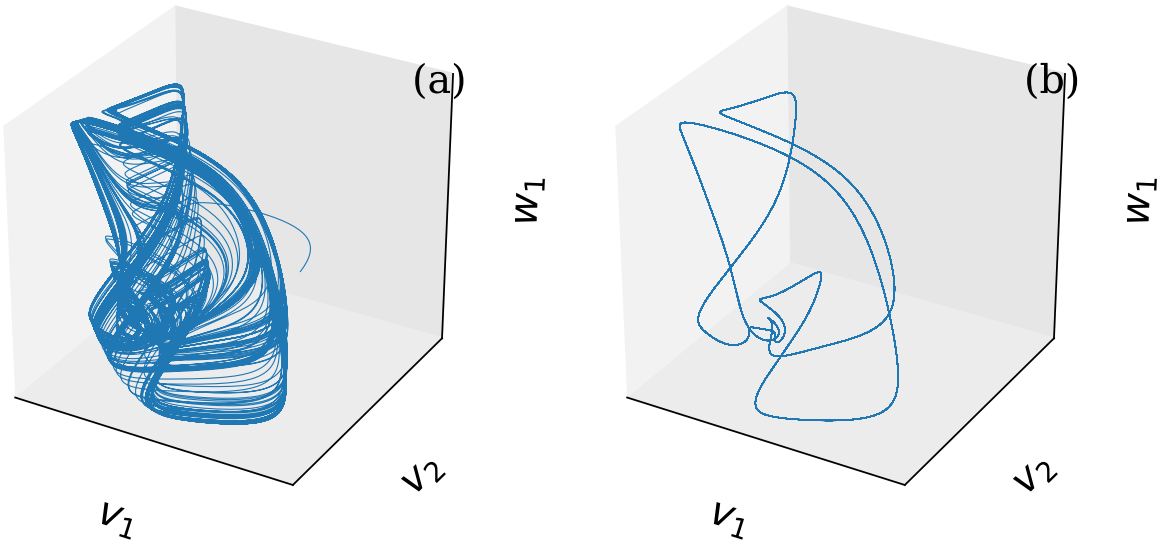

Figure 2.6: Three dimensional view, v_1, v_2, and w_1 of the (a) *Initial* ($0 \leq t \leq 2000$) and (b) *Final* ($9600.91 < t \leq 12000$) state for Eq 2.8, for initial conditions: $(v_1, w_1, g_1, v_2, w_2, g_2) = (0.5, 0.3, 0.96, 0.8, 0.1, 0.96)$. Parameters: $(a, b, t_s, A, \omega) = (8, 2.5, 2000, 0.0217, 2\pi f)$ with $f = 0.1271$.

each other by a half period. A sample of each neuron's time series is given in Figure 2.8. Performing a cross-correlation of the last 1000 seconds, or 100000 samples, of the *Final* state we find that v_1 differs from v_2 by a half period of 1364 samples.

Another way of presenting the results is given in Figure 2.9, which displays a "peak map" of both the *Initial* and *Final* states, where we plot ordered pairs of peak values (p_i, p_{i+1}) for i running through all of the peaks in $v_1(t)$. So, the horizontal axis represents the initial peak value, p_i. The vertical axis represents the value of the peak immediately following, p_{i+1}. For a periodic system having only five maxima, such as when $g = 0.82$, the peak map would only show five data points. With a chaotic system, since the peaks do not occur periodically, the peak map will have many distinct data points, as seen in Figure 2.9a. Here, the *Final* state for v_1 is shown to be a period five time series, as seen in Figure 2.9b where only 5 blue dots appear. Since the two signals are only out of phase, but otherwise are identical, v_2 is also a period five signal with the same maxima as shown by the red circles in Figure 2.9b.

To summarize the results, in Figure 2.10 we present an analogous procedure to that

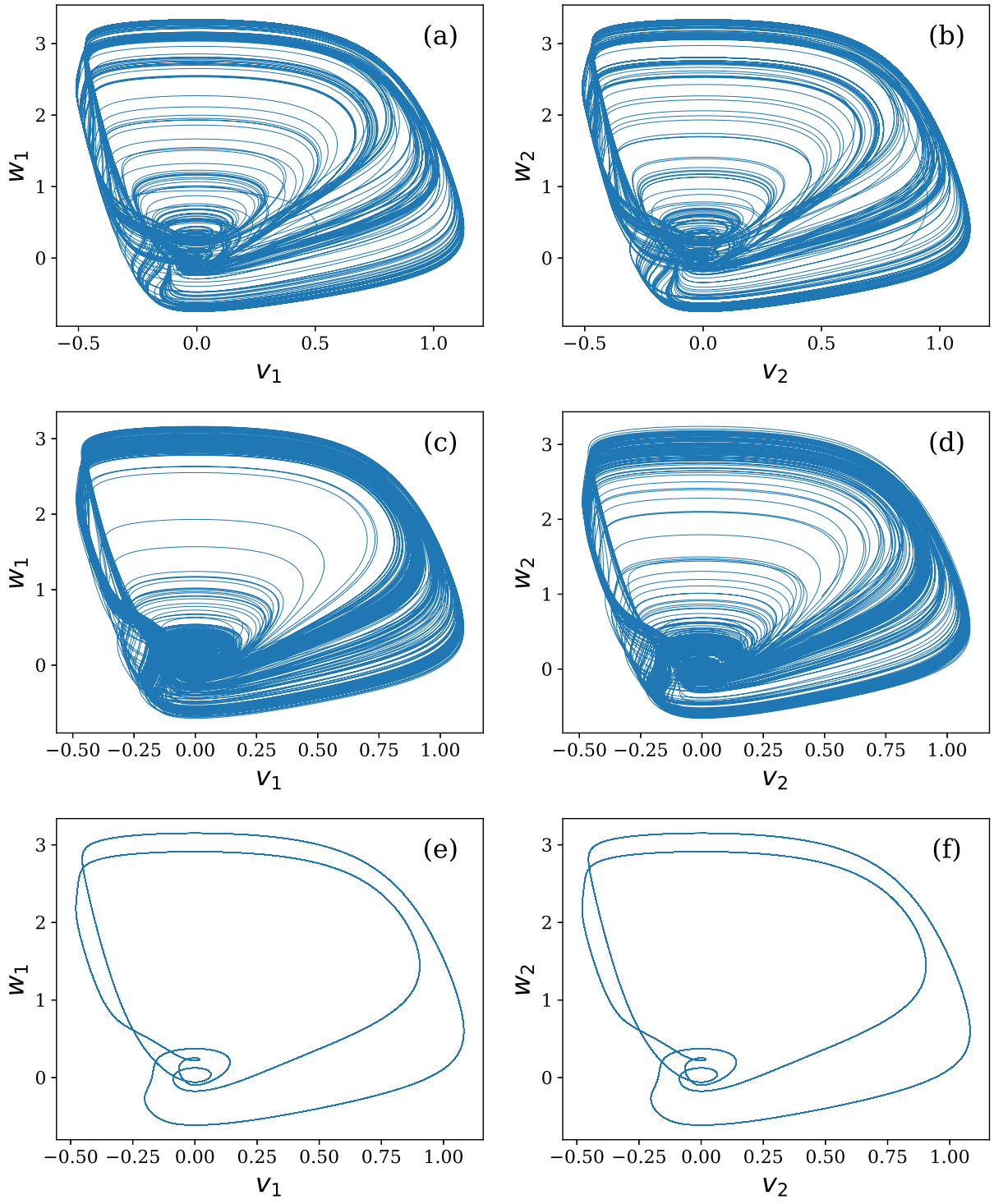

Figure 2.7: Phase space of N1 and N2 in the *Initial*, *Transient*, and *Final* states. (a) N1 *Initial* state; (b) N2 *Initial* state; (c) N1 *Transient* state; (d) N2 *Transient* state; (e) N1 *Final* state; (f) N2 *Final* state.

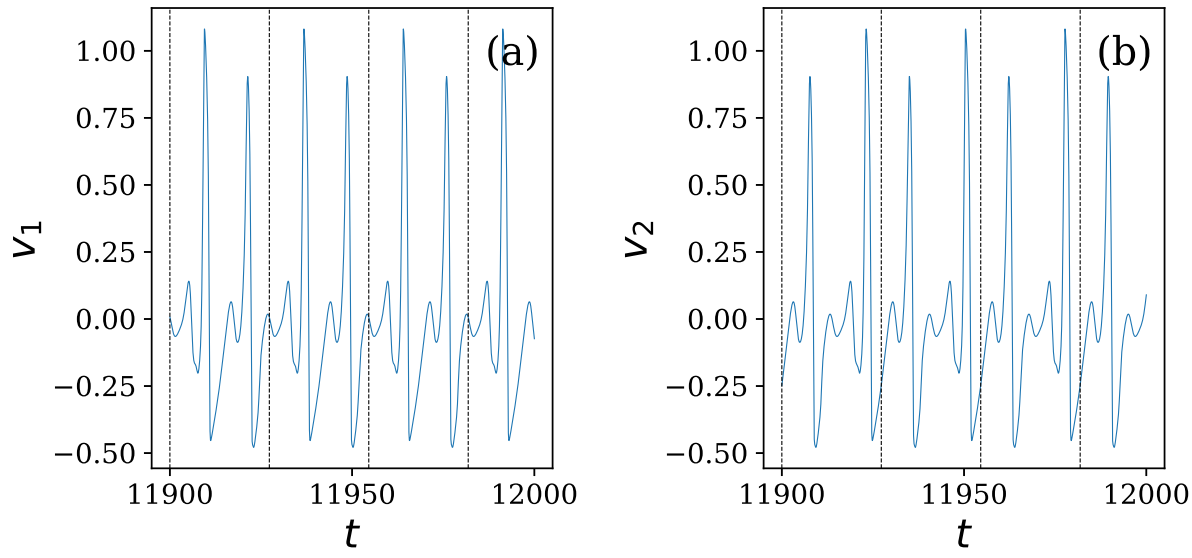

Figure 2.8: Final 100 seconds of v_1 and v_2 *Final* states. Period for each is 27.28 (simulated) seconds or 2728 samples. The sampling rate is at 100 samples/second. (a) Neuron 1 v_1 time series. (b) Neuron 2 v_2 time series.

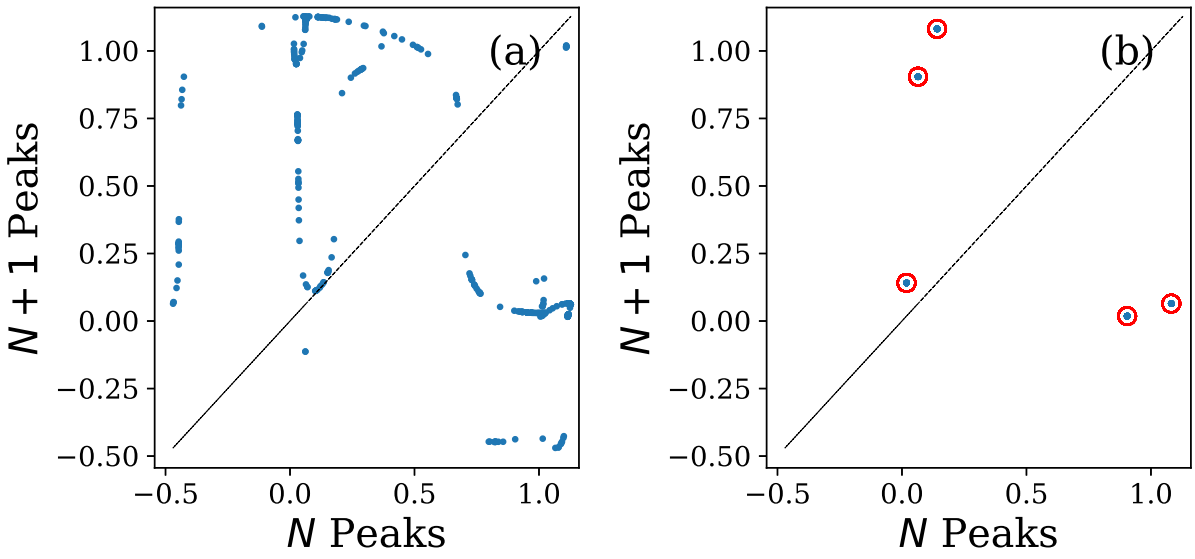

Figure 2.9: (a) Peak-Next Peak map for *Initial* state of v_1. (b) Peak-Next Peak map for *Final* state of v_1 (blue dots) and v_2 (red circles).

presented in Figure 1.1. In Figure 2.10a we begin with two neurons evolving chaotically with bidirectional interaction. The introduction of $I_0(t)$ to N1 plays the role of an external control. This control induces a change in the dynamics of the system. Figure 2.10b shows the arrival of the external signal, $I_0(t)$, at N1. As the signal begins to impact N1, we see in Figure 2.10c that the dynamics of N1 begin to change and become less chaotic, shown by scars forming as the thick outer rings. Since the communication between the two neurons is moderated by the sigmoidal synaptic strength, N2 is being driven by the less chaotic dynamics of N1 as it passes through the transient phase. This is shown in Figure 2.10d as N2 begins to be driven towards a similarly less chaotic state. In Figure 2.10e N1 has locked into a period 5 periodic state with N2 still transitioning. However, once the signal becomes exponentially negligible in Figure 2.10f we see that both N1 and N2 are in a periodic state that is period 5. This is because the interaction function has replaced the role of the external signal, $I_0(t)$, as the driver towards a periodic state. Instead, N1 continually drives N2 into a periodic state and N2 drives N1 into a periodic state. Therefore the chaotic neurons have now locked into a state of mutual stabilization.

The analogous interaction function for the coupled neurons is the synaptic strength, g, which has become a function that evolves by synaptic learning. We see that the mutual stabilization of chaotic systems presented in Morena & Short [4] and Morena & Short [7], and Short & Morena [8] has an analogous mechanism that can be applied to a chaotic neural system as depicted in Figure 2.10.

2.7 Discussion

Chaotic systems communicating through an interaction function have the capability to exhibit periodic behavior through mutual stabilization [3, 5, 6]. Previous work took a more information theoretic approach, but we have established mutual stabilization using feedback between neurons and a saturating sigmoidal learning effect that allows the synaptic strength to evolve dynamically.

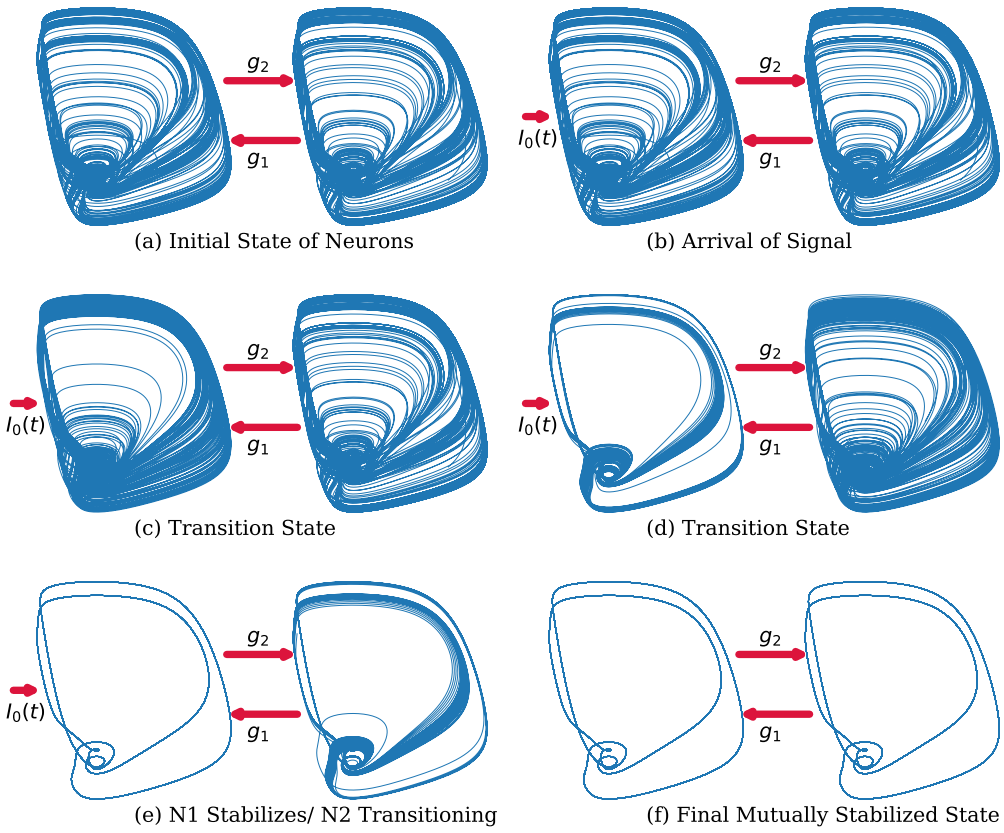

Figure 2.10: Diagrams outlining the dynamics of mutual stabilization between two interacting chaotic FitzHugh-Nagumo neurons as given in Eq 2.8. (a) N1 and N2 evolving in the *Initial* state. (b) External signal $I_0(t)$ reaches N1. (c) N1 enters *Transient* state while $I_0(t)$ is still active. (d) Both N1 and N2 are in the *Transient* state. (e) N1 has reached a stable *Final* state and N2 is in the *Transient* state. (f) N1 and N2 reach a mutually stabilized *Final* state, at which point the external signal, $I_0(t)$, is negligible.

As seen in the bifurcation diagram, Figure 2.4, the bidirectional FHN model has both chaotic and periodic regions that naturally arise for different values of the synaptic strength. By allowing the parameter g to become a dynamical variable, the coupled FHN model is allowed the ability to evolve from a chaotic to a periodic state and if the periodic state occurs where the synaptic strength has saturated, the periodic behavior will continue to persist. The transition from chaotic to periodic behaviors is mediated by the dynamics of g, which exhibit oscillatory learning with a sigmoidal envelope. It has been shown that an appropriate choice of $I_0(t)$ (and the parameters A and ω) and the targeted final saturation level for g_f produces the desired transition. The specific dynamical model of the synaptic strength does not appear to matter as long as it has appropriate saturation characteristics. In future work, we intend to explore this in more detail. The primary result shown here is that through feedback and sigmoidal learning it is possible that coupled neurons in an initially chaotic state may undergo mutual stabilization into a phase locked, persistent periodic state.

One potential natural extension of this work is the investigation of how autaptic synapses may be involved to induce stabilization. Autaptic synapses are when the axon of a neuron connects back to its own dendrites. Recent experimental work has shown that the effects of autapses on inhibitory neural firing are stronger than that of traditional inhibitory synapses [34], whereas recent theoretical work with an autaptic HH neuron demonstrated spontaneous spiking [35]. This may allow modification of our model where one neuron may receive direct feedback from its own output that is transmitted back to its dendrites from the axon in addition to the synaptic input from the other neuron.

The mutual stabilization established in this work focused on coupling through the variable representing membrane voltage, v. However, in Tessone *et al.* [36] it was shown that oscillatory suppression can occur through the inhibitory variables, w, of FHN neurons. If the model in Tessone *et al.* [36] can exhibit chaotic and periodic behavior as the coupling parameter in the w variable varied then it is possible that mutual stabilization may arise. Other future work will focus on expanding upon this finding in relation to variations of this model,

other types of incoming signals, and other neural models. In the Hindmarsh-Rose model the firing dynamics of a single neuron is constructed using three dynamical variables. This means the individual neuron has the capability to be chaotic, similar to the double scroll oscillator in §1.2. This implies a more direct connection to the mechanism in Figure 1.1 than in Figure 2.10. Specifically, the remaining chapters of this dissertation will investigate coupling between chaotic Hindmarsh-Rose neural models where input to a connected neuron is modulated by a spike-like external signal.

Furthermore, since Figure 2.4 indicates numerous regions of periodic behavior, many different mutual stabilization states exist. If neurons are mutually stabilizing after receiving a stimulus like an external neural signal, then the persistent state has "stored" something about the incoming signal. We intend to explore the concept of stored information of the external signal in the following chapters.

The content of this chapter has been adapted and reproduced from Parker, J. E. & Short, K. M. Sigmoidal synaptic learning produces mutual stabilization in chaotic FitzHugh-Nagumo model. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **30,** 063108 (June 2020), with the permission of AIP Publishing.

CHAPTER 3
EXISTENCE OF CUPOLETS IN THE HINDMARSH-ROSE MODEL

3.1 Introduction

Throughout recent decades, many approaches have been taken to try to understand biological neurons. In this chapter, we provide insight into neurons through examining the Hindmarsh-Rose (HR) mathematical model of neural firing. The HR neuron is represented by a set of coupled nonlinear ordinary differential equations, Eq 3.1. This model is particularly interesting for mathematical investigations because it is 3-dimensional and can exhibit periodic, quasi-periodic, and chaotic behavior. The development of the HR model is outlined below.

Previously discussed in Chapter 1, the seminal work of Hodgkin and Huxley in 1952 provided foundational insight connecting biology and the mathematics of an individual neuron. Many alternative models have since been proposed that better reflect the complex dynamics of the HH model. One of the first was the FitzHugh-Nagumo (FHN) model [13, 14] and was the focus of Chapter 2. The FHN model is a simplification of the HH model from 4-dimensions to 2-dimensions that is capable of exhibiting a single spike or periodic spiking. Even though the FHN model exhibits similar firing dynamics as the HH model, the parameters in the FHN model are not directly related to the biological or chemical properties of the neuron. Instead, one variable reflects the membrane potential and the other variable reflects the recovery dynamics. Another 2-dimensional neural system, called the Morris-Lecar model [37], is also capable of similar dynamics as the FHN model, but with parameters that are more closely connected to the biological and chemical properties of the

neuron. In 1985, Chay [38] presented a 3-dimensional neural model as a reduced HH system similar to the Morris-Lecar model where the parameters reflect biophysical properties. The dynamics of this model are further explored in Fan & Chay [12]. An extension of the FHN model was introduced by Hindmarsh and Rose as a 3-dimensional autonomous dynamical system, now known as the Hindmarsh-Rose (HR) model [39]. Similar to the FHN model, the HR model parameters are not biochemically based, yet the overall dynamics are consistent with observed neural firing patterns. One key benefit of the added dimension of the HR model is that it enables the possibility for chaotic dynamics, which will be the focus of the work reported here. For a reference on many commonly used mathematical neural models, see Izhikevich [40].

Chaotic dynamics occur when the behavior of a dynamical system is aperiodic, remains bounded for all time, and is sensitive to initial conditions. Often this is characterized by a positive exponential rate of divergence between two close trajectories, commonly measured by the maximal Lyapunov exponent (MLE). In particular, the HH model is capable of exhibiting chaotic dynamics and an example of chaotic dynamics within the original parameters is discussed in Guckenheimer & Oliva [41]. Rabinovich & Abarbanel [42] suggest that chaotic behavior is important for connected networks of neurons as a means to produce desired rhythms, citing lobster central pattern generators (CPG) as an example. The authors then show how a network of HR neurons can model a CPG even though each individual neuron would have chaotic dynamics without the coupling. Faure & Korn [43] and Korn & Faure [44] discuss the role of chaos within the brain. It is evident that understanding the interactions between chaotic neurons is essential. Erichsen *et al.* [45] discuss periodic and chaotic dynamics that occur when two HR neurons are coupled. Shuai & Durand [26] explore phase synchronization between coupled HR neurons. Recently, Doungmo Goufo & Tabi [46] reported on the influence of an external current on the Hindmarsh-Rose system, focusing on numerical investigation into chaotic poles of attraction.

In this chapter, we further examine the role of chaos in the brain and, specifically, we

examine how a single chaotic neuron can produce many periodic states dependent on received inputs. In particular, we discuss a control scheme applied to a chaotic HR neuron that causes the system to stabilize onto periodic orbits. These stabilized periodic orbits are highly accurate unstable periodic orbits. The unstable periodic orbits of a chaotic system create a skeleton that a chaotic trajectory will traverse when evolving [47].

The control scheme, adapted from Hayes *et al.* [9, 10] and described in detail in Parker [3], Zarringhalam [5], Zarringhalam & Short [6], Morena & Short [7], Short & Morena [8], Morena [48], Morena *et al.* [49], and Morena & Short [50], uses a repeated binary control sequence (e.g. 101100) to apply relatively small perturbations or controls along a control plane. There are two types of controls designated by the selected bit in the control sequence, microcontrols (0) and macrocontrols (1). The control planes are constructed as a hyperplane transversal to the attractor trajectories, referred to as a Poincaré section. Each bit of a binary control sequence, called the control bit, is taken in sequence as a trajectory intersects a control plane. When the trajectory intersects with a control plane and the control bit is a 0, only a tiny perturbation called a microcontrol is applied, as described in more detail below. When the control bit is a 1, a larger, predetermined perturbation called a macrocontrol is applied along the control plane. This adapted control scheme has been applied to other dynamical systems, mainly the double scroll oscillator [51]. When certain control sequences are applied to a chaotic system via this control scheme, the chaotic behavior stabilizes into persistent, periodic dynamical structures called cupolets (*c*haotic, *u*nstable, *p*eriodic, *o*rbit-*lets*). These cupolets were originally described in Parker [3] and discussed with applications in more detail in Zarringhalam [5], Zarringhalam & Short [6], Morena & Short [7], Short & Morena [8], Morena [48], Morena *et al.* [49], Morena & Short [50], and Short *et al.* [52, 53]. In this chapter, we will demonstrate how this control scheme stabilizes chaotic HR behavior onto periodic, persistent cupolets. In §3.2 we discuss the chaotic HR neural model and provide a visualization of the chaotic regime of interest. Then, in §3.3 we further explain the control sequence used to produce cupolets and illustrate properties of the system that

result from applying controls. We conclude in §3.4 by providing examples of certain cupolets and discuss implications of this work in §3.5.

3.2 Model

The chaotic neural system of interest is the Hindmarsh-Rose model. Eq 3.1 provides the original HR model as given in Hindmarsh & Rose [39]. The original parameter values used are $a = 1$, $b = 3$, $c = 1$, $d = 5$, $s = 4$, $x_r = -8/5$, with r and I varying. The parameter I represents a direct current input to the dimensionless membrane potential, x. The membrane potential continuously absorbs input until firing an action potential and then enters a refractory period. The dimensionless y and z variables represent the recovery dynamics governing the refractory period.

$$\begin{aligned} \dot{x} &= y - ax^3 + bx^2 + I - z \\ \dot{y} &= c - dx^2 - y \\ \dot{z} &= r(s(x - x_1) - z) \end{aligned} \tag{3.1}$$

The z recovery variable has a smaller timescale than the y recovery variable due to the value of the r parameter. Hindmarsh & Rose [39] discusses the different dynamics Eq 3.1 can exhibit. Typically, Eq 3.1 exhibits bursting behavior often seen in biological neural observations. That is, the x variable has multiple consecutive spikes (action potentials) before entering the refractory period governed by the recovery dynamics (y and z variables). This bursting behavior cannot occur in the previously discussed FitzHugh-Nagumo neural model. Hindmarsh & Rose [39] report random bursting with $r = 0.005$ and $I = 3.25$. We focus on a similar parameter set that produces chaotic dynamics where $r = 0.006$ and $I = 3.25$.

This parameter set was selected after first conducting a bifurcation analysis. This allowed for examination of the different types of long-term spiking behavior (periodic or chaotic) with different I values. In Figure 3.1, a bifurcation diagram shows the different behaviors for Eq

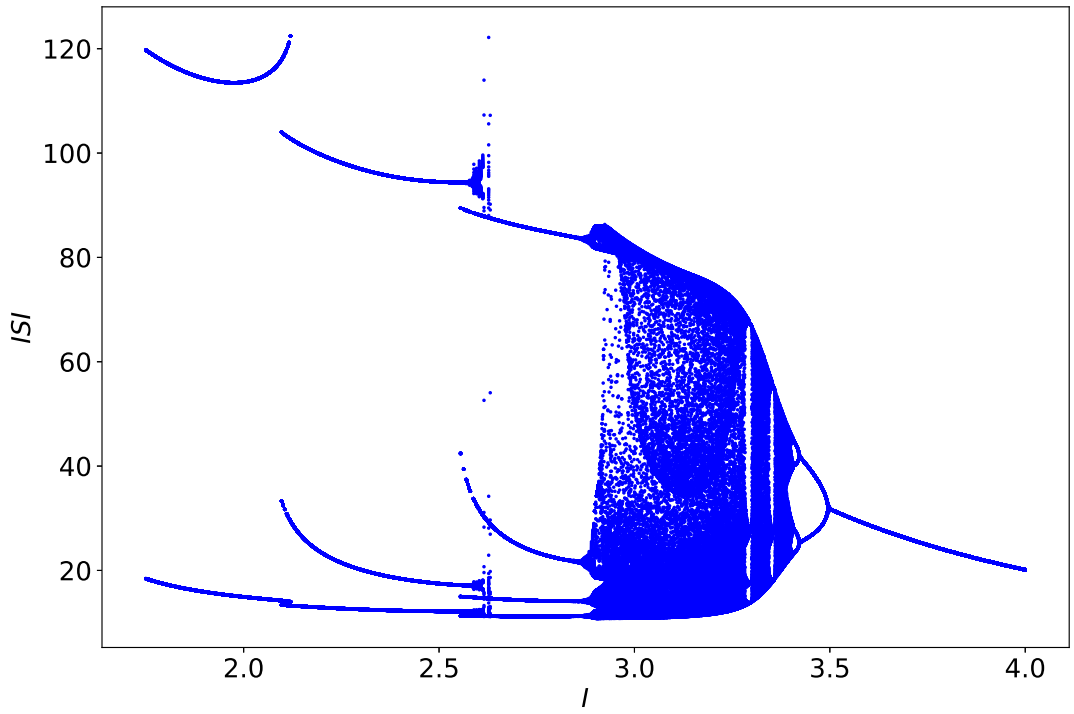

Figure 3.1: Bifurcation diagram for the HR model in Equation 3.1. The bifurcation parameter, I, is on the horizontal axis and varies from $I_{min} = 1.750$ to $I_{max} = 4.000$, with $\Delta I = 0.001$. The vertical axis shows the interspike interval (ISI) of the x variable in Eq 3.1.

3.1 as I is varied. The bifurcation parameter, I, is on the horizontal axis ranging from $I_{min} = 1.750$ to $I_{max} = 4.000$ with a $\Delta I = 10^{-3}$. The interspike interval (ISI) is on the vertical axis and measures the time between two consecutive spikes. At a given I value on the horizontal axis, $I = I_0$, a thick band of ISI values would indicate that Eq 3.1 evolves chaotically (e.g. $I_0 = 3.25$), whereas if there are only a few ISI values plotted above a given I value, then that indicates regular, periodic behavior. For example, in Figure 3.1 when $I < 2.5$ and $I > 3.5$ the HR model produces periodic behavior. Innocenti *et al.* [54] provides a more detailed analysis of the possible dynamics exhibited by varying I and further explores chaotic dynamics of the HR neuron in Innocenti & Genesio [55]. Storace *et al.* [56] analyze the bifurcations that occur when varying the two parameters I and b.

From Figure 3.1, it can be seen that $I = 3.25$ leads to chaotic behavior of the HR model.

Figure 3.2 shows the phase space resulting from numerical integration of Eq 3.1 with these parameters. Figure 3.3 shows the chaotic time series. Figure 3.3A shows the x time series where the number of consecutive spikes before entering a refractory period, called a burst, does not have any pattern and consists of 2, 3, or 4 spikes. Figure 3.3B shows the y time series, and Figure 3.3C shows the z time series. Neither Figure 3.2 or Figure 3.3 contain periodic behavior.

All simulations in this chapter were performed numerically using an explicit Runge-Kutta 4th order method with a $dt = 1/128$. In Figure 3.2 and Figure 3.3 the total simulation time of $t_{final} = 10000$ gives 1280000 iterations, which provides enough time for the system to settle onto the attractor after any initial transient.

3.3 Control Scheme

In this section we detail the control scheme for the HR neural system. It is adapted from the control scheme for the double scroll oscillator developed by Hayes *et al.* [9, 10] that was used to generate cupolets in Parker [3], Zarringhalam [5], Zarringhalam & Short [6], Morena & Short [7], Short & Morena [8], Morena [48], Morena *et al.* [49], and Morena & Short [50]. This adapted control scheme uses two control planes that exist on different sections of the attractor that represent two different stages of neural firing. Controls are applied only when the trajectory intersects with a control plane. The control planes are lower dimensional (two-dimensional) Poincaré sections that are approximately orthogonal to the flow of the trajectory.

The control planes are defined after numerical integration of Eq 3.1 following an initial transient. The location of the control planes are chosen such that each control plane reflects an aspect of the bursting behavior of the system. One control plane, called Poincaré section 0 (PS0), is defined as the plane corresponding to the average local minima of the x time series of each spike. This results in a value of $x = -0.9832605683131186$. This plane encompasses the refractory dynamics of the neural system that occurs after firing a burst of spikes. The

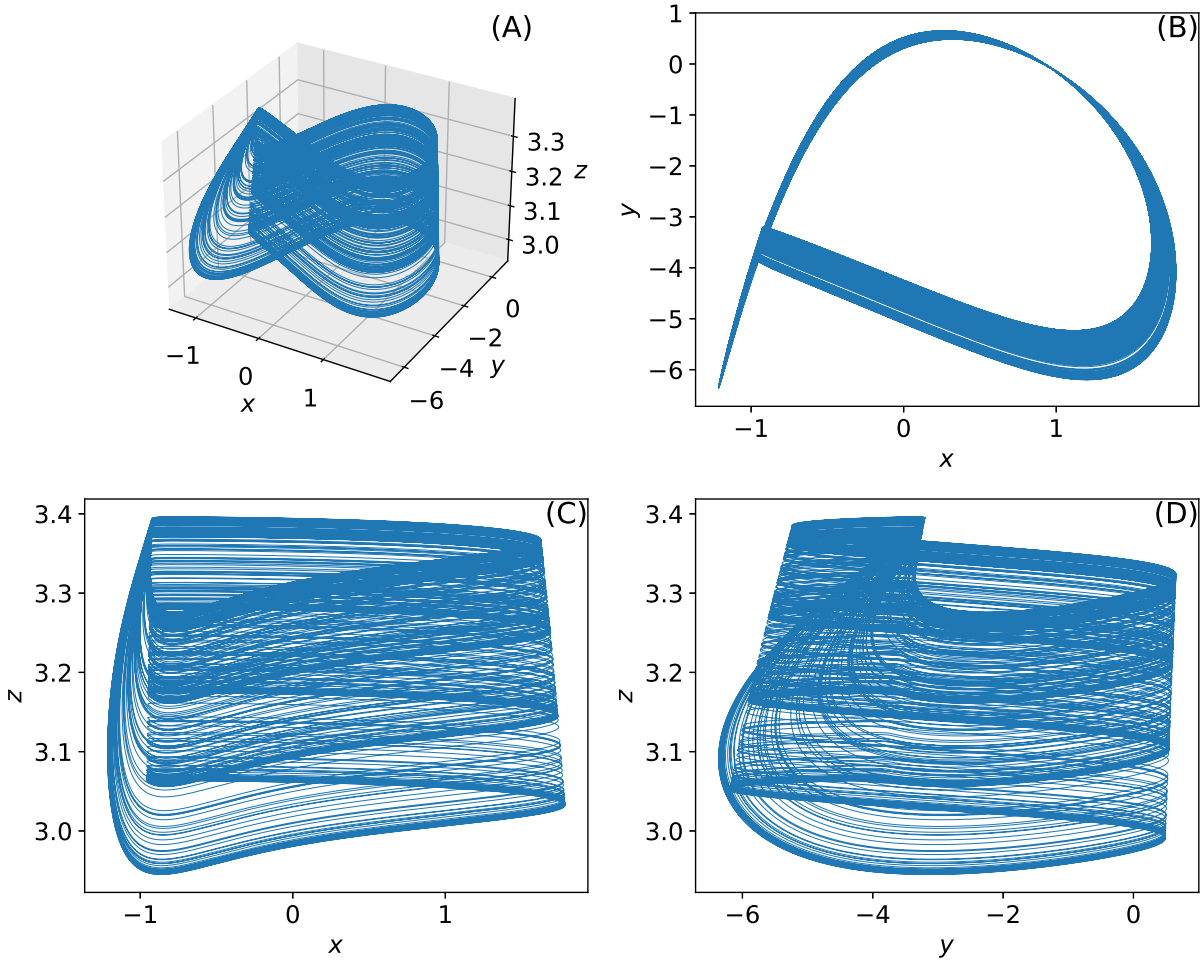

Figure 3.2: Numerical integration of Eq 3.1 with the original parameters, $a = 1$, $b = 3$, $c = 1$, $d = 5$, $s = 4$, $x_r = -8/5$, $r = 0.006$, and $I = 3.25$. Numerical integration was performed using the explicit Runge-Kutta 4th order method with a fixed timestep of $dt = 1/128$ for 1280000 iterations. The final 960000 iterations (75% of numerical integrations) are displayed showing that any transient behavior has passed. (A) 3-Dimensional phase space plot of x, y, and z dynamics. (B) Projection of (A) into the x-y plane. (C) Projection of (A) into the x-z plane. (D) Projection of (A) into the y-z plane.

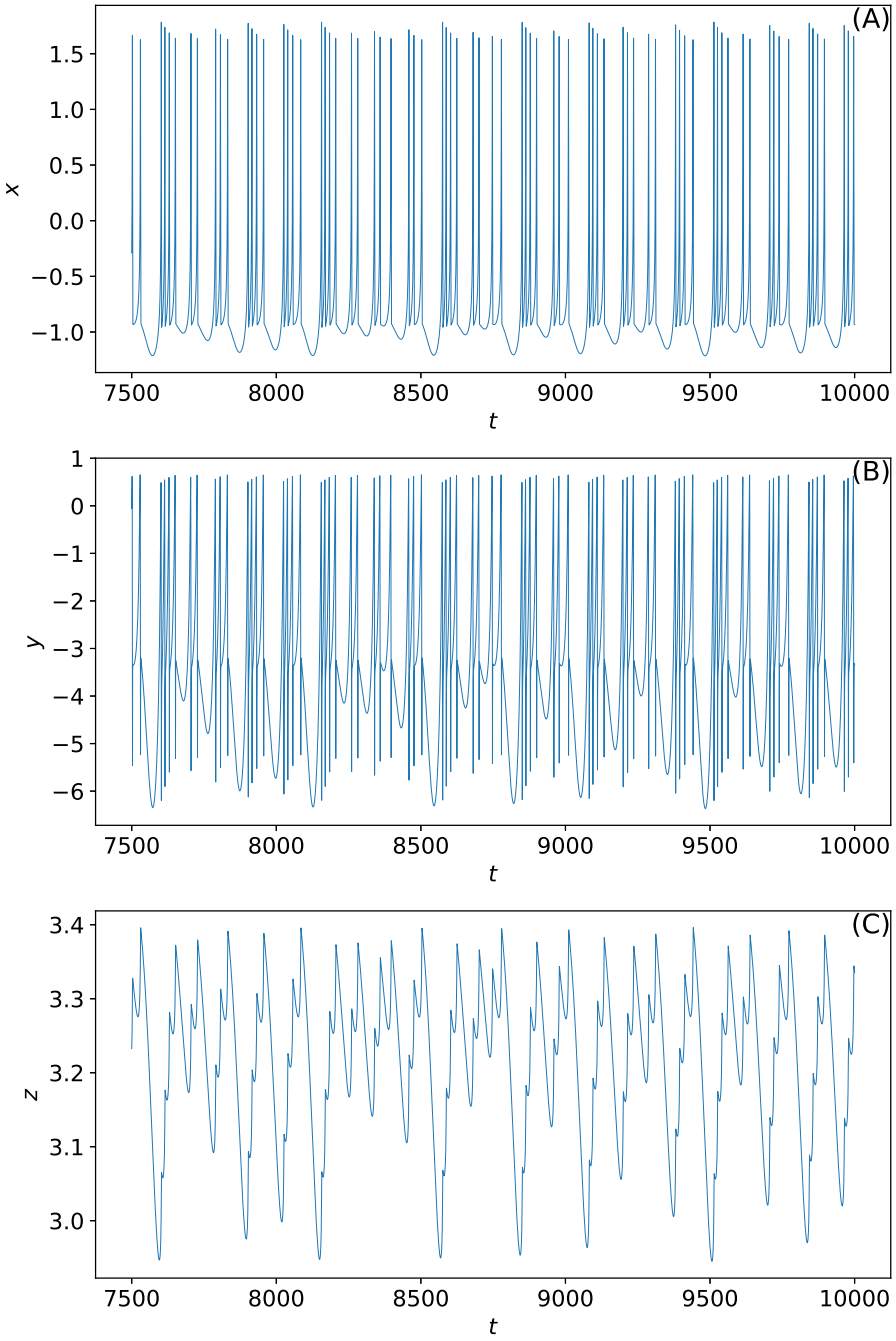

Figure 3.3: Numerical integration of Eq 3.1 with the original parameters, $a = 1$, $b = 3$, $c = 1$, $d = 5$, $s = 4$, $x_r = -8/5$, $r = 0.006$, and $I = 3.25$. Numerical integration was performed using explicit the Runge-Kutta 4th order method with a fixed timestep of $dt = 1/128$ for 1280000 iterations. The final 320000 iterations (25% of numerical integrations) are shown. (A) x time series. (B) y time series. (C) z time series.

second control plane, called Poincaré section 1 (PS1), is defined as the plane corresponding to the average y value of the average x local maxima, resulting in $y = -3.3657609537434663$. After numerical integration of Eq 3.1 to determine the control planes, Henon's trick [57] is used to find the exact points of intersection of the trajectories with each plane. Henon's trick multiplies a given dynamical system, $\frac{d\vec{x}}{dt}$, by the reciprocal of the corresponding dynamics of the variable that defines the plane (e.g. $\frac{d\vec{x}}{dt}\frac{dt}{dy} = \frac{d\vec{x}}{dy}$ for PS1). The point exactly on the plane can be identified by first locating the point immediately preceding the plane and then integrating the distance from that point to the control plane, a distance dx for PS0 or a distance dy for PS1. Figure 3.4 shows each control plane (outlined as a black rectangle) and the corresponding exact points on each plane (yellow for PS0 on the left and purple for PS1 on the right).

A line of best fit for the points on each control plane is calculated to give a function that approximates the points on the plane, $p_0(\sigma = y)$ for PS0 and $p_1(\sigma = x)$ for PS1. The polynomial allows each control plane to be reduced to 1-dimension. That is, $p_i(\sigma)$, for $i = 0, 1$ allows for reconstruction of every point on the control plane from the independent variable (x for PS0 and y for PS1). In Figure 3.5A the resultant line of best fit (degree 2 polynomial) is given where the yellow points are directly on the control plane PS0. The low residual squared (1.71×10^{-33}) indicates that $p_0(\sigma = y)$ is an excellent approximation to return a z value for a given $\sigma = x$ value on the plane. In Figure 3.5B the resultant line of best fit (degree 3 polynomial) is given where the purple points are directly on the control plane PS1. The low residual squared (7.16×10^{-15}) indicates that $p_1(\sigma = x)$ is a very good approximation to return a z value for a given $\sigma = y$ value on the plane.

Each control plane is then partitioned into $M = 1600$ equally spaced bins via the independent variable. The x values of PS0 are partitioned into M bins and the y values of PS1 are partitioned into M bins. The middle of each bin is used as an initial condition of a trajectory and the the binary sequence that a trajectory visits is recorded, called the symbolic dynamics, where a 0 indicates the trajectory visits PS0 and a 1 indicates the trajectory visits

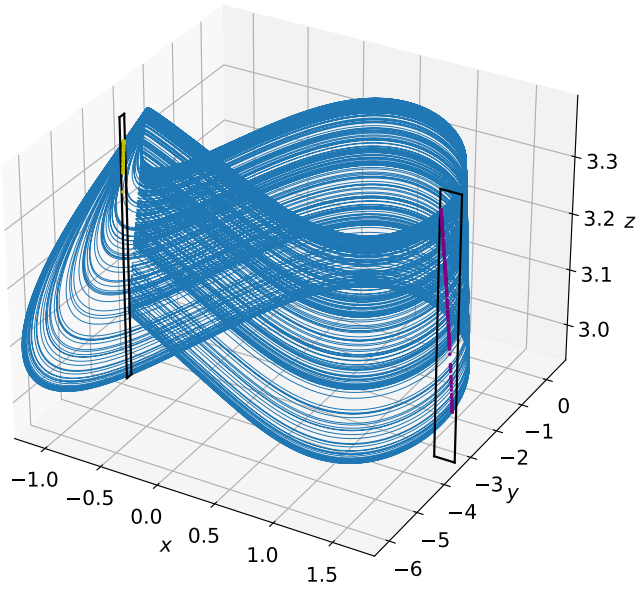

Figure 3.4: Control planes of Eq 3.1 after numerical integration. Each control plane is outlined as a black rectangle. The left control plane is PS0 and is defined as the plane corresponding to $x = -0.9832605683131186$. The yellow points inside PS0 are the points found exactly on the plane using Henon's trick [57]. The right control plane is PS1 and is defined as the plane corresponding to $y = -3.3657609537434663$. The purple points inside PS1 are the points found exactly on the plane using Henon's trick.

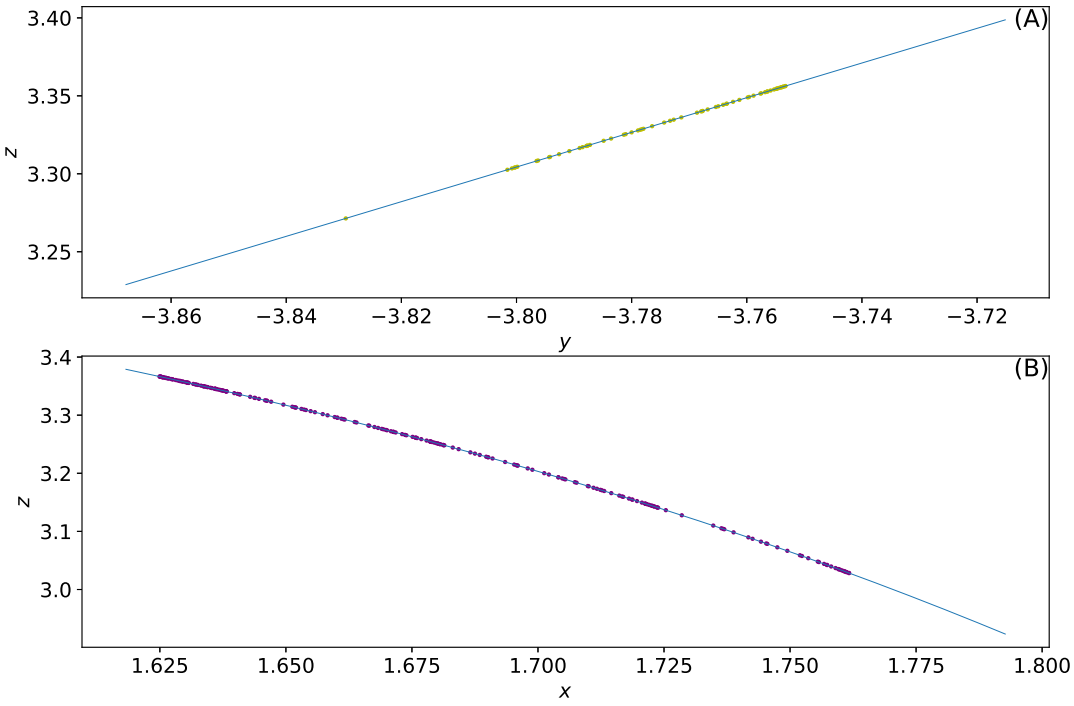

Figure 3.5: Polynomial fit of the points on each Poincaré section. (A) Blue line is the degree 2 polynomial fit, $p_0(\sigma = y)$, of the yellow points (residual squared is 1.71×10^{-33}). The yellow points are points found from integrating directly onto PS0 using Henon's trick. The polynomial returns a z value for a given $\sigma = y$ input. (B) Blue line is the degree 3 polynomial fit, $p_1(\sigma = x)$, of the purple points (residual squared is 7.16×10^{-15}). The purple points are points found from integrating directly onto PS1 using Henon's trick. The polynomial returns a z value for a given $\sigma = x$ input.

PS1. The symbolic dynamics are generated for N future iterations around the attractor. As an example, if $N = 4$ the symbolic sequence of a bin B on PS0 might be 1011. This sequence means the numerical integration from B results in a trajectory crossing PS1 (1), then PS0 (0), then PS1 twice (11). Each N-bit long binary sequence is mapped as a base-10 real number. The coding function generates this number and is given as $r_N(X) = \sum_{n=1}^{N} b_n 2^{-n}$, where $b_1 b_2 b_3 ...$ represents the binary symbolic dynamics [9]. Figure 3.6 provides a visualization of the coding function with $M = 1600$ bins and $N = 16$ future intersections with the control planes. Each plateau represents identical symbolic dynamics, meaning that for 16 loops around the attractor a trajectory starting in a bin on the plateau visits the same sequence of control planes. Figure 3.6A plots the PS0 coding function $r_N(X)$ on the vertical axis with the bin number on the horizontal axis. Figure 3.6B plots the PS1 coding function $r_N(X)$ on the vertical axis with the bin number on the horizontal axis. Figure 3.6 is distinctly different from the coding function of the double scroll oscillator (see Parker [3], Zarringhalam [5], Morena & Short [7], Hayes *et al.* [9], Morena *et al.* [49], and Morena & Short [50]) that contains many jagged lines and few plateaus.

After establishing the coding function and bin centers for each control plane, a binary control string, $CTRL$, can be applied. This sequence, $CTRL$, is repeated until a cupolet is formed or the simulation time ends. Each bit of $CTRL$ is iterated through cyclically when the trajectory intersects with one of the control planes. Once the trajectory intersects with a control plane, the bin that the trajectory has entered is identified. The designated control that is applied to the trajectory corresponds to the current bit in $CTRL$. Two types of controls can occur when the trajectory intersects with a control plane: (i) microcontrols and (ii) macrocontrols. If the current bit is 0 then a microcontrol is applied. A microcontrol recenters the trajectory to the middle of the bin so the shift is bounded to have a magnitude smaller than $(1/1600)L$, where L is the length of the Poincaré section. If the bit is a 1 then a macrocontrol is applied. A macrocontrol perturbs the trajectory to a new bin center that has the smallest difference in symbolic sequence when compared to the symbolic sequence

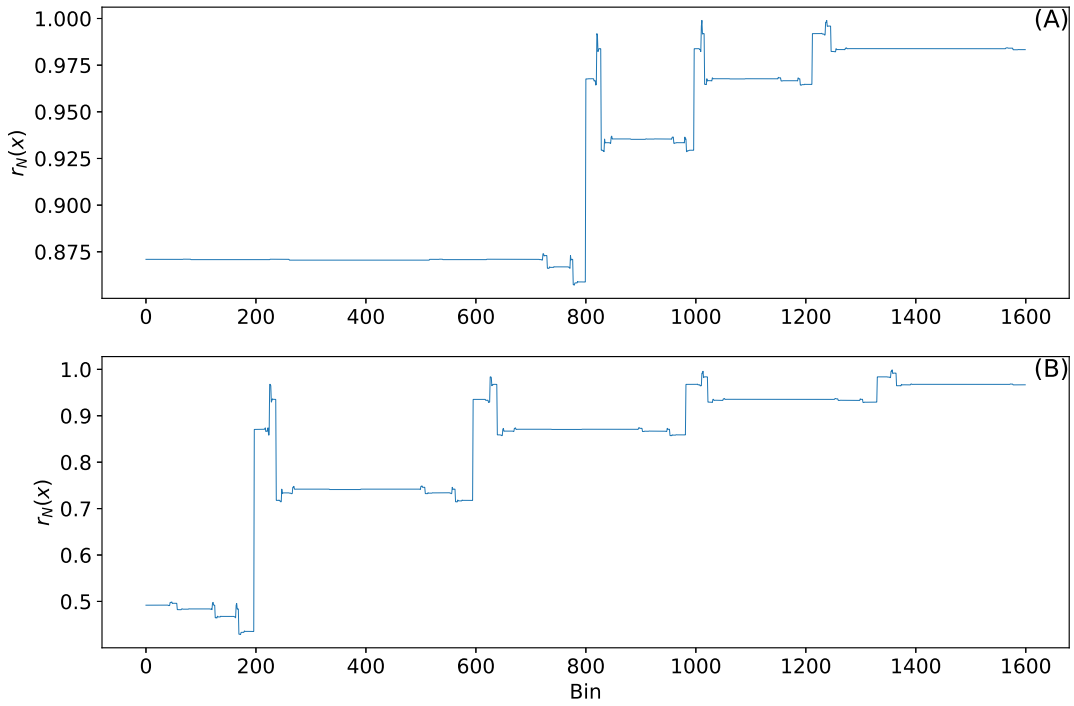

Figure 3.6: Coding function $r_N(X)$ with $M = 1600$ bins and $N = 16$ future intersections with the control planes. (A) PS0 coding function where the input x represents the bin center along $p_0(x)$. (B) PS1 coding function where the input x represents the bin center along $p_1(y)$. The plateaus represent identical symbolic dynamics while the jagged peaks are bins with different symbolic dynamics.

of the current bin center. For $N = 16$, this corresponds to a perturbation from bin x_0 to a new bin x_1 where that minimizes the difference in coding function, $|r_N(x_0) - r_N(x_1)|$.

Due to the bursting nature of the Hindmarsh-Rose dynamics, constraints exist on the symbolic dynamics. The uncontrolled symbolic dynamics result in one or more consecutive intersections of PS1 but never more than one consecutive intersection with PS0. This leads to certain interactions between the bins in the control planes and is illustrated by Figure 3.7.

Figure 3.7A and B show the resultant evolution from each bin center to the next intersection of a control plane. The horizontal axis is the initial condition on the control plane (PS0 bins in Figure 3.7A, PS1 bins in Figure 3.7B). The vertical axis is the corresponding control plane and bin first encountered after numerical integration. In Figure 3.7A, if a given point (p, q) is colored in purple, it means that a trajectory beginning at the center of bin p on PS0 results in a next intersection with bin q on PS1, and since all bins on PS0 encounter PS1 next, we see only purple dots in the Figure 3.7. If a point (p, q) in Figure 3.7B is colored yellow, it indicates that a trajectory beginning at the center of bin p on PS1 results in an intersection with bin q on PS0 if yellow, but if the point is colored purple if the next intersection is with bin q on PS1. Figure 3.7C and D show the inverse map of Figure 3.7A and B, respectively. That is, a point (p, q) in Figure 3.7C indicates that a trajectory starting at bin q (colored yellow if q is on PS0 and purple if q is on PS1) results in the next intersection of bin p PS0. A point (p, q) in Figure 3.7D indicates that a trajectory starting at bin q (colored yellow if q is on PS0 and purple if q is on PS1) results in the next intersection of bin p PS1. Each bin on PS1 evolves to only one bin, but many trajectories from either PS0 or PS1 result in the multiple next intersection with the same bin on PS1. Every bin on PS0 has an next intersection with bin on PS1. PS0 bins do not feed back into PS0 bins; only PS1 bins feeds into PS0 bins. However, PS1 bins can feed back into PS1 bins. This mapping governs what control sequences will result in a cupolet.

Figure 3.7 provides insight for a mathematical definition of a cupolet. Each control plane

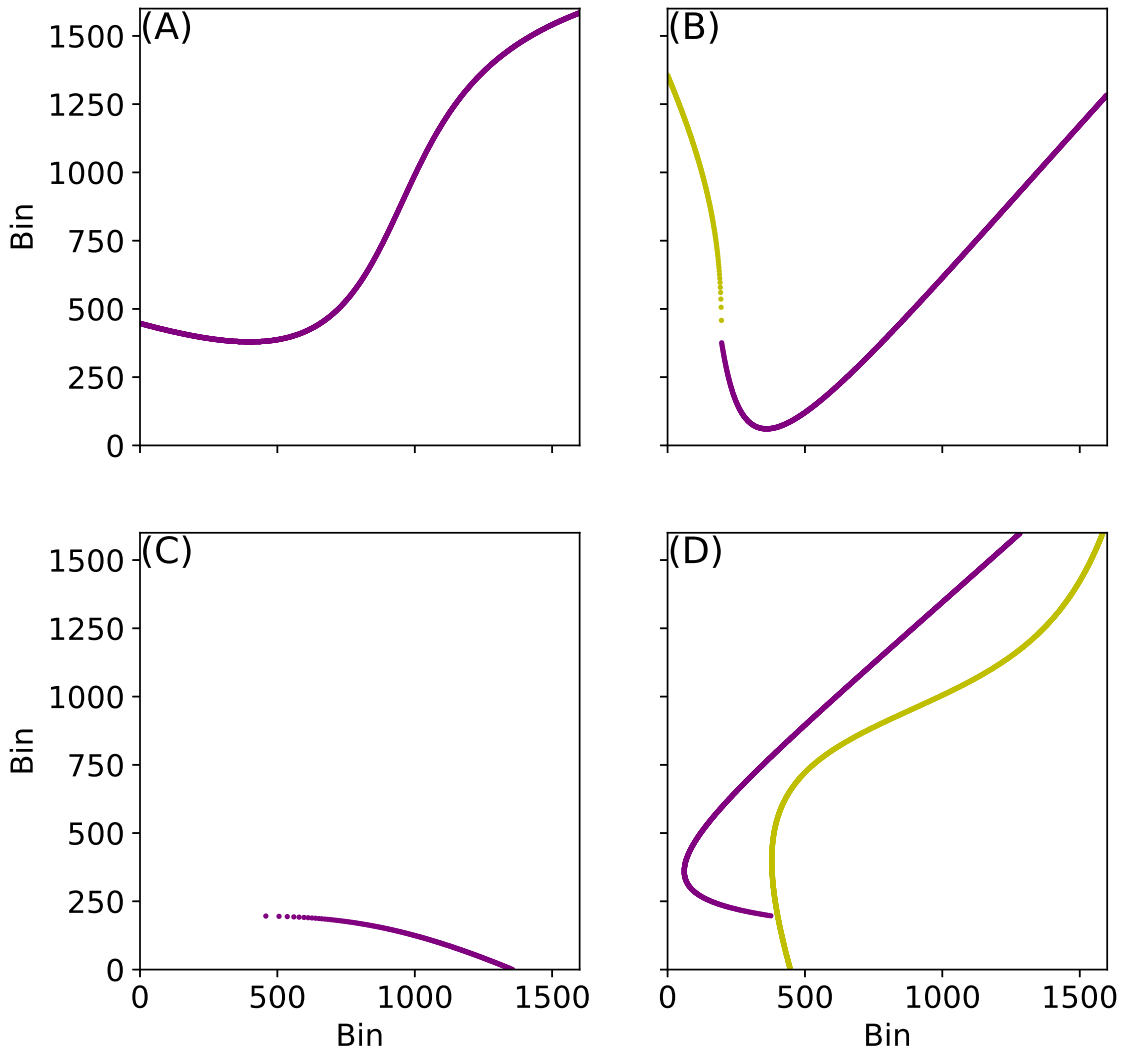

Figure 3.7: Maps of interacting bins between PS0 and PS1. PS0 bins are indicated by yellow points and PS1 bins are indicated by purple points. (A) First return map for PS0, where the horizontal axis indicates the bin centers along PS0. The vertical points are the first intersecting bin and control plane of the trajectories that start at the corresponding horizontal bin, with the color indicating the plane. (B) First return map for PS1, where the horizontal axis and vertical axis are analogous to (A). (C) Map of every trajectory that starts at a bin center that maps into a bin on PS0. The horizontal axis represents the bin centers along PS0. The vertical points are the trajectories that start at the corresponding bin center that evolves towards the PS0 bin centers on the horizontal axis, where the color indicates the plane of origin. (D) Map of every trajectory that starts at a bin center that maps into a bin on PS1, where the horizontal axis and vertical axis are analogous to (C).

consists of M bins and from Figure 3.7 the interactions between the bins on each control plane are clear. Consider the PS0 bins as a set Φ and the PS1 bins as a set Δ. A cupolet then would be the resulting controlled trajectory that visits the same sequence of bins for each bit in the control sequence. For example, take the control sequence 001. Let δ_i be the bin on Δ and the first control bit is a 0. Let this bin, δ_i, be a bin that evolves to a new bin δ_j on Δ and the next control bit is another 0. After applying another microcontrol, this δ_j then evolves to a new bin, and a new plane, ϕ_k on Φ. However, the next control bit is a 1 so a macrocontrol is applied. The macrocontrol perturbs from ϕ_k to ϕ_r on Φ. The new bin ϕ_r maps to δ_i on Δ. At this point, the next control bit resets to the beginning of the control sequence. Since the current bin is the same as the initial bin, δ_i, and the control sequence has restarted, we know that this is a cupolet. An example of this cupolet is provided in Figure 3.8. With proper identification of the different interactions between each control plane, coupled with the mapping provided by the macrocontrols, one can identify cupolets without performing any numerical integration.

In the next section, this control scheme is applied to Eq 3.1 and selected cupolets are analyzed.

3.4 Cupolets

Several cupolets and cupolet properties are discussed in this section using $M = 1600$ bins and $N = 16$ for $r_N(X)$. Binary control sequences from 2 bits to 8 bits are used. This results in over 500 control sequences applied to the Hindmarsh-Rose system. Not every control sequence will result in a cupolet. A cupolet is considered to have formed when the control sequence leads to stabilization of the chaotic trajectory into a persistent, periodic orbit that would otherwise not exist without the control since the system is in a chaotic state.

If a particular control sequence does result in a cupolet, then it is named with a 'C' followed by the control sequence. For example, C10 would correspond to the cupolet generated with the control sequence 10. Each cupolet also has a visitation sequence that is the

binary string representing the symbolic dynamics of the cupolet. The symbolic dynamics reflect the order of control planes that the cupolet visits and is designated by placing a 'V' prior to the symbolic dynamics. For example, cupolet C01011 has a visitation sequence of V110111101111011. Due to the implemented control sequence 01011, repeated cyclically, the chaotic Hindmarsh-Rose has collapsed into a persistent, periodic sequence of three four-spike bursts. Another example is given in Figure 3.8, cupolet C001, that is one of the simplest cupolets.

Several more cupolets are given in Figure 3.9. These cupolets are generated from different control sequences. Figure 3.9A is C11, Figure 3.9B is C0110, Figure 3.9C is C1010010 and Figure 3.9D is C01010010. Figure 3.10 further examines these cupolets by displaying the time series of the x variable, representing the firing pattern of the neuron and the corresponding spike raster plot. A spike raster plot provides the time for when a particular event occurs. In this case, the event is the peak of a spike or action potential. Spike raster plots are a convenient way to visualize timing of events. Figures 3.10A-B shows one period of C11. This cupolet has 6 bursts before repeating where the number of spikes in the burst are 2, 3, 4, 3, 3, 4. Figures 3.10C-D shows one period of C0110. This cupolet has 3 bursts before repeating where the number of spikes in the burst are 4, 3, 2. Figures 3.10E-F shows one period of C1010010. This cupolet has 3 bursts before repeating where the number of spikes in the burst are 3, 4, 4. Figures 3.10G-H shows one period of C01010010. This cupolet has 4 bursts before repeating where the number of spikes in the burst are 2, 3, 3, 4. Even though these cupolets are created through control of a chaotic HR neuron, the bursting pattern is not the same. Table 3.1 provides an overview of the patterns seen in each cupolet in Figure 3.9. The column labeled "Spikes per Period" is the integer number of spikes that occur over one period, or the number of events representing spikes in each spike raster plot from Figure 3.10. For instance, C11 contains 19 spikes in one period while C0110 has 9 spikes. C1010010 has 11 spikes while C01010010 has 12 spikes in one period. The approximate period of each cupolet has been rounded to two decimal places. The approximate spikes per unit time, or

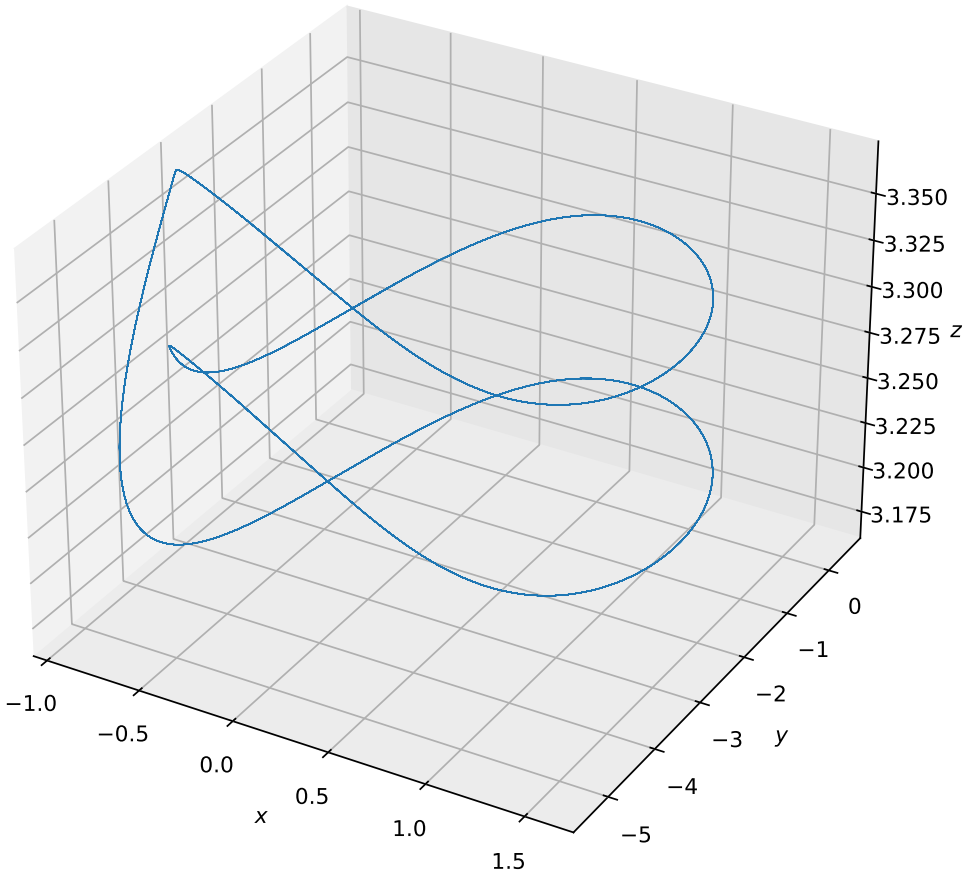

Figure 3.8: Cupolet C001.

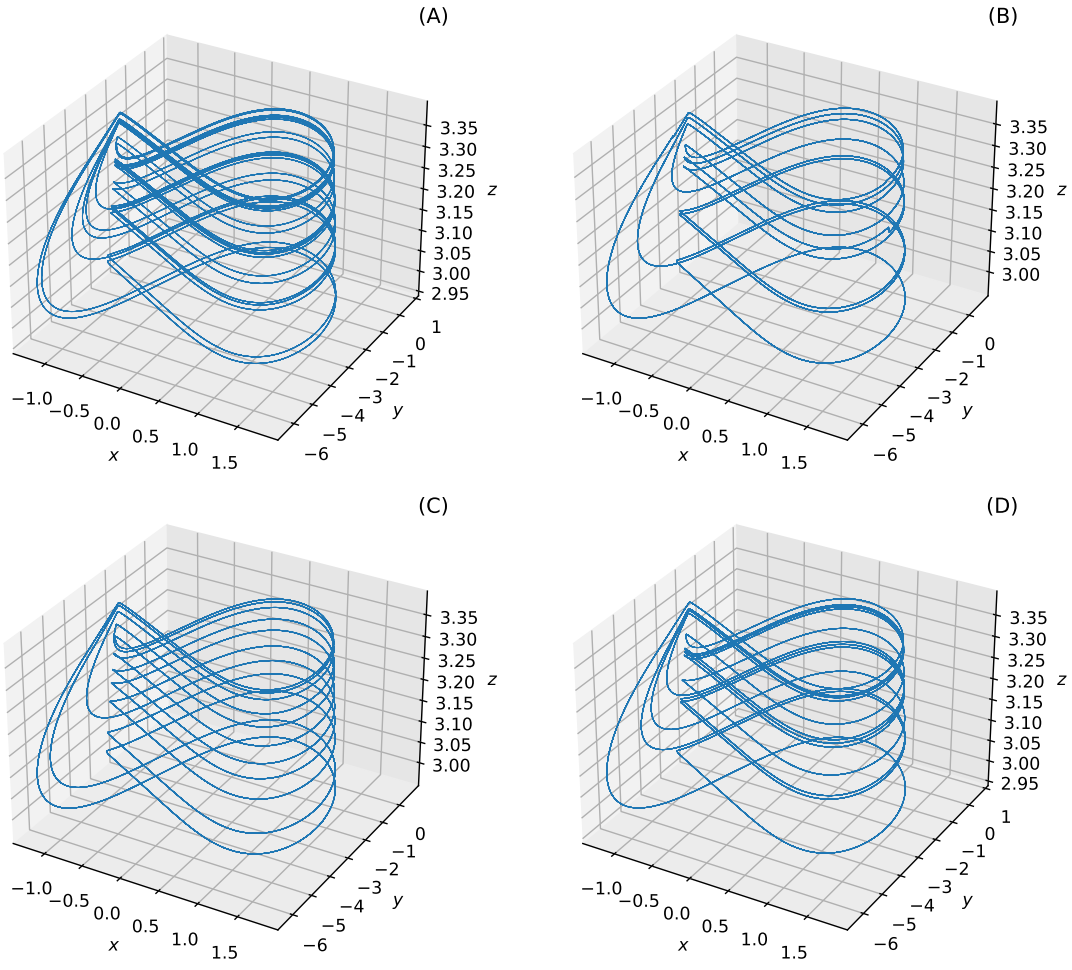

Figure 3.9: (A) Cupolet C11. (B) Cupolet C0110. (C) Cupolet C1010010. (D) Cupolet C01010010.

Cupolet	2 Bursts	3 Bursts	4 Bursts	Spikes	T,Period Length	Spikes/T
C11	1	3	2	19	639.43	0.03
C0110	1	1	1	9	301.25	0.03
C1010010	0	1	2	11	360.96	0.03
C01010010	1	2	1	12	406.15	0.03

Table 3.1: Summarized characteristics of a single period of the cupolets in Figure 3.9. The "bursts" column headings indicate the number of spikes in a burst event, and the entries in the columns indicate the number of such burst events within a single period of the cupolet in the row label. The period length, T, is measured in integration time units and has been rounded to two decimal places. The spikes/T, or firing rate, is ratio of spikes to the period length and has been rounded to two decimal places.

firing rate, is reported in the last column of this table rounded to two decimal places. This collection of cupolets has a wide range of differences in both spikes and period, yet the firing rate remains approximately the same.

Certain control sequences generate a multiplicity of cupolets, also called homologous cupolets. That is, for a given control sequence it is possible that more than one cupolet can be generated. For a given control sequence with homologous cupolets, the specific cupolet generated depends on the initial bin and control plane where the first control in the sequence is applied. For example, C11 has two homologous cupolets meaning that two unique cupolets result from applying the control sequence 11 repeatedly. The homologous cupolets for C11 are given in Figure 3.11. Figure 3.11A shows cupolet C11A and Figure 3.11B shows cupolet C11B. C11A and C11B have distinctly different orbits and spiking behavior. This can be further seen by examining Figure 3.12, where the respective spike raster plots of one period for each cupolet.

Figure 3.12 provides the spike raster plot of one period of the x time series for each C11 type. C11A has fewer spikes (10 spikes) than C11B (19 spikes) in a single period. C11A has a shorter period (271.42 time units) compared to C11B (639.43 time units). The approximate spikes/period are 0.04 for C11A and 0.03 for C11B.

Figure 3.13 illustrates which bins from PS1 will result in C11A (blue) and C11B (red). Starting in each bin of PS1, the control sequence 11 is applied until a cupolet forms. Then

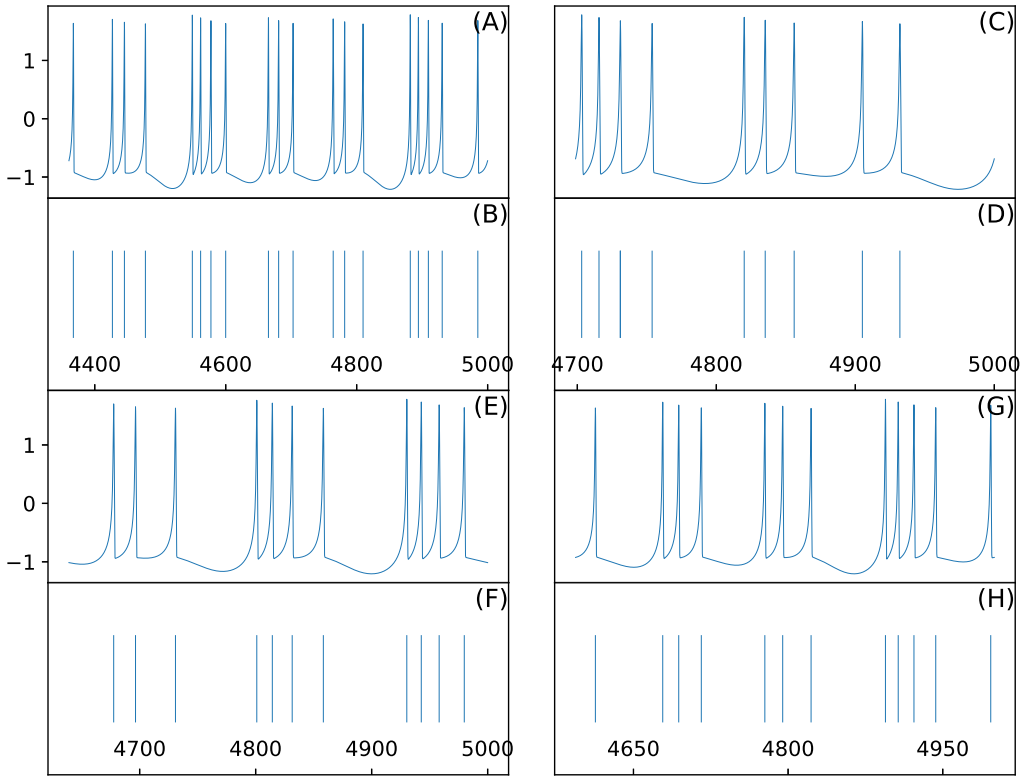

Figure 3.10: (A-B) Cupolet C11 single period x time series (A) and corresponding spike raster plot (B). (C-D) Cupolet C0110 single period x time series (C) and corresponding spike raster plot (D). (E-F) Cupolet C1010010 single period x time series (E) and corresponding spike raster plot (F). (G-H) Cupolet C01010010 single period x time series (G) and corresponding spike raster plot (H).

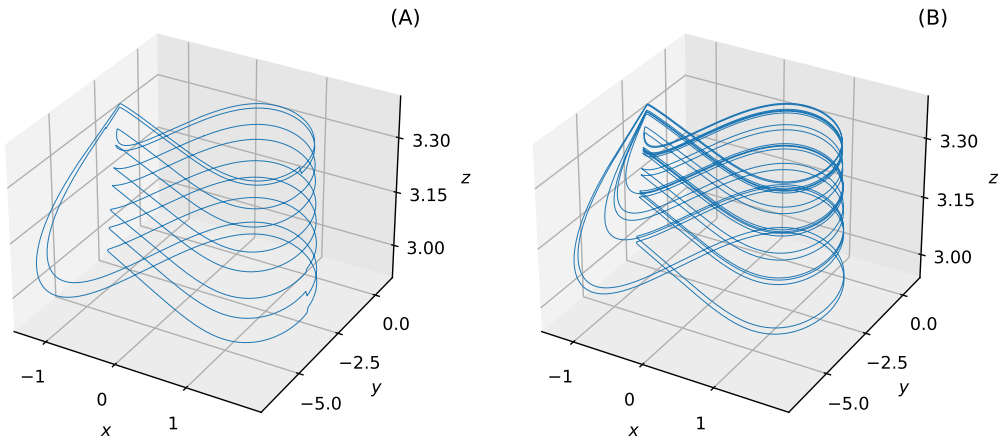

Figure 3.11: Homologous cupolets of C11. (A) First type of C11 cupolet, designated C11A. (B) Second type of C11 cupolet, designated C11B.

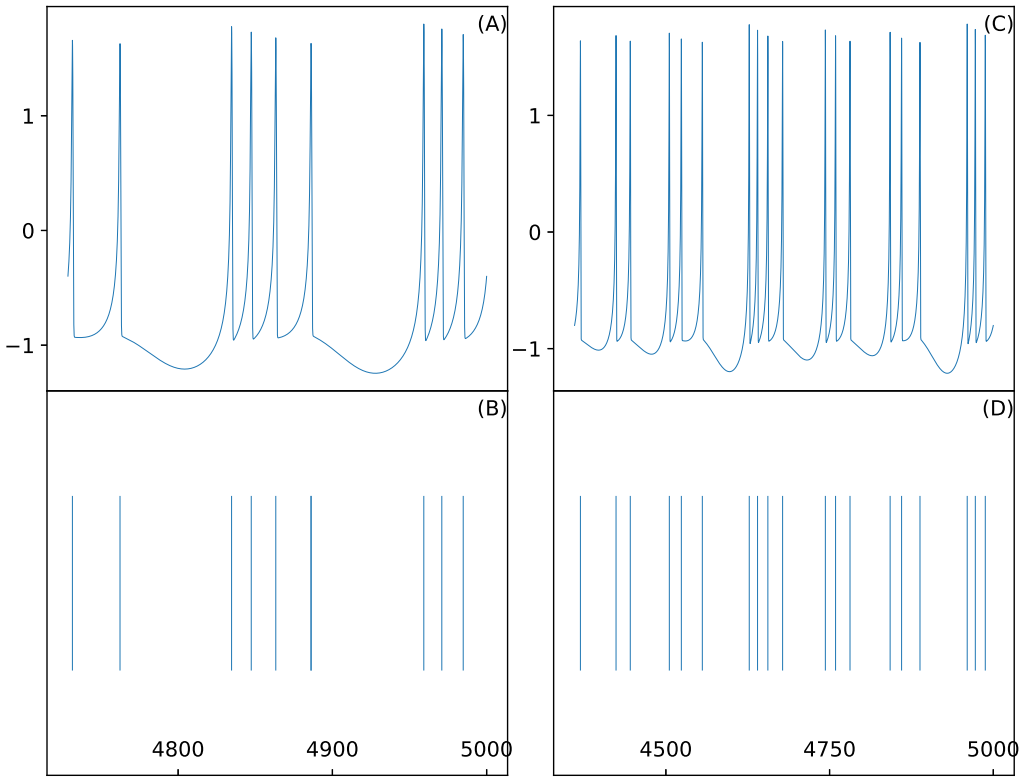

Figure 3.12: (A-B) Type C11A single period x time series (A) and corresponding spike raster plot (B). (C-D) Type C11B single period x time series (C) and corresponding spike raster plot (D).

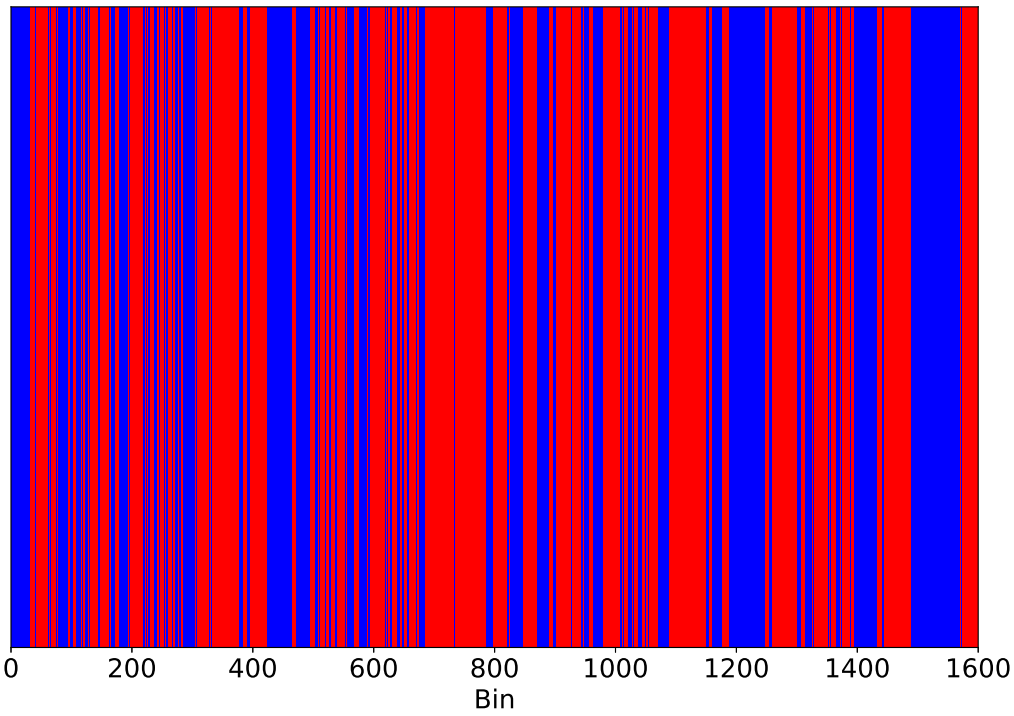

Figure 3.13: Graphical depiction of the distribution of initial bins that stabilize onto C11A and C11B. Blue (Red) bar indicates that starting the control sequence in the corresponding bin on the horizontal axis resulted in C11A (C11B).

the type of cupolet is checked and if the cupolet is C11A then a blue bar of height 1 is plotted above the corresponding bin number on the horizontal axis. If the cupolet is C11B then a red bar of height 1 is plotted above the corresponding bin number on the horizontal axis. C11A resulted from 42.625% of the bins, or 682 bins while C11B resulted from 57.375% of the bins, or 918 bins.

In addition to these cupolets and by using the mapping approach described in §3.3, 8127 cupolets have been found so far with the control sequence lengths up to a total of 12 bits. Several more examples of cupolets are given in Figure 3.14 and the properties of these cupolets are given in Table 3.2.

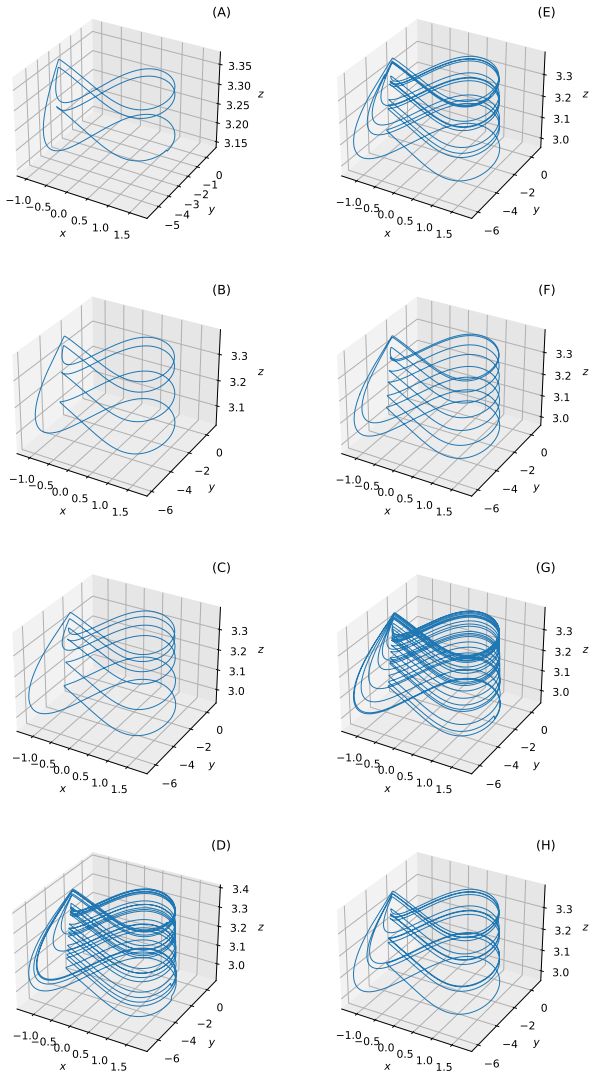

Figure 3.14: Various cupolets resulting from the control scheme. (A) C01000100, (B) C10000, (C) C11010011, (D) C1101110, (E) C11100010, (F) C10010, (G) C0111110, (H) C01100011.

Cupolet	Spikes	T,Period Length	Spikes/T
C01000100	3	116.29	0.03
C10000	4	137.20	0.03
C11010011	6	196.67	0.03
C1101110	25	697.27	0.04
C11100010	12	408.53	0.03
C10010	8	257.52	0.03
C0111110	22	739.86	0.03
C01100011	12	404.05	0.03

Table 3.2: Summarized characteristics of a single period of the cupolets seen in Figure 3.14. The spikes are the total number of spikes, or events in the raster plot, per period. The period length, T, is measured in integration time units and has been rounded to two decimal places. The spikes/T, or firing rate, is ratio of spikes to the period length, T and has been rounded to two decimal places.

3.5 Discussion

Rabinovich & Abarbanel [42] suggest a role for chaos in the brain and how certain individual neurons in a CPG would exhibit chaotic dynamics if the neuron was not connected. We consider the implications and possible dynamical evolution that might occur if some neurons are in a mathematical parameter state that would exhibit chaotic behavior when no inputs are being received. However, as shown here, the many inputs within the nervous system may stabilize the chaos into a regular pattern. The different binary control sequences act as a analog for myriad biological inputs a neuron might receive. In §3.4, we saw that some binary control sequence cause the neuron to stabilize onto a cupolet. However, we have found that not all control sequences will result in a cupolet and some control sequences result in homologous cupolets. Therefore, a neuron may have many unique behaviors, corresponding to unique cupolets, yet the system parameters remain unchanged and the HR neuron would be chaotic without the stabilization induced by the controls. The controls are convenient "stand-ins" or representation of myriad incoming spike-like signals that be input into a neuron.

A complete breakdown and investigation into the full space of accessible cupolets and

the relationships between the cupolets is suggested, but is beyond the scope of this work. Morena & Short [50] provide an analysis of the cupolets from the double scroll system to the extent where certain cupolets are shown to be fundamental. These fundamental cupolets can then be used to generate a composite cupolet. Isolating the fundamental cupolets within the chaotic Hindmarsh-Rose system may provide further insight into the potential dynamics of the system and properties of neural dynamics.

In this chapter we reported on the existence of cupolets within a chaotic neural system, the Hindmarsh-Rose dynamical model. Even though there are no direct biological parameters in the model, the dimensionless x variable represents observed neural firing. Izhikevich [40] discusses the usefulness of certain neural models and reports the capability of the Hindmarsh-Rose model to generate many observed neural firing patterns. This suggests that cupolets may exist biologically, where the controls are applied through auxiliary cells or even connected neurons. Morena & Short [7] demonstrated how interacting cupolets in the double scroll oscillator can mutually stabilize through an interaction function. In mutual stabilization, the signal of the first cupolet is used as the input to the interaction function. The output of the interaction function is the control on the second cupolet, then the process is repeated from the second cupolet back to the first cupolet. For certain interaction functions, and certain cupolets, this communication results in the two cupolets locking into a periodic, persistent interaction with no external controls. One interaction function used in Morena & Short [7] was based on integrate-and-fire neural dynamics. This interaction function motivated the result in Parker & Short [18]. The authors reported how two bidirectional neurons, each modeled as an adapted FitzHugh-Nagumo model [13, 14], were capable of transitioning to a periodic state from a chaotic state in the presence of a certain external signal. This suggests a bidirectional Hindmarsh-Rose model may be able to achieve the mutual stabilization shown in Morena & Short [7] and Parker & Short [18]. The advantage of the Hindmarsh-Rose model over the FitzHugh-Nagumo model is that each individual neuron can exhibit chaotic behavior leading to a more direct comparison with the mutual stabiliza-

tion shown in Morena & Short [7]. In the next chapter, we examine the proper mechanism in which communicating Hindmarsh-Rose neurons can achieve mutual stabilization.

We note this is an initial report of cupolets in a chaotic Hindmarsh-Rose system. It is reasonable to hypothesize that cupolets exist in a higher dimensional neural model capable of chaos or other neural models, for example the Hodgkin-Huxley model which has direct biological parameters. A similar control scheme to the control scheme that has generated cupolets here has not been attempted on a chaotic system with 4-dimensions, although the generation of cupolets should be possible.

CHAPTER 4
INFORMATION THEORETIC NEURAL MUTUAL STABILIZATION

4.1 Introduction

In Chapter 1, we discussed an overview of published work on mutual stabilization in communicating double scroll oscillators and provided Figure 1.1 for illustration. In Figure 1.1, and reported in Morena & Short [7], mutual stabilization was shown to exist between interacting cupolets of the double scroll oscillator. The authors demonstrated that there exist certain interaction functions that transform the visitation sequence of one cupolet into a control sequence on the second cupolet. In mutual stabilization, this process repeats from the second cupolet back to the first cupolet, where the interaction function transforms the visitation sequence of the second cupolet into a control on the first cupolet. For certain interaction functions and certain cupolets, the cupolets would lock into a state of persistent, periodic behavior, called mutual stabilization, as long as the interaction persisted. Morena & Short [7] detailed various interaction functions that resulted in mutual stabilization between two double scroll oscillators. In this chapter, we apply a similar process to communicating Hindmarsh-Rose neurons to see if mutual stabilization can occur. The communication between the neurons is mediated by an interaction interaction function called *Integrate-And-Fire* that was used in Morena & Short [7]. This chapter first discusses the *Integrate-And-Fire* interaction function and the reasoning for the choice of this type of interaction. Then we show the results of a feed-forward or unidirectional system and a feedback or bidirectional system of communicating Hindmarsh-Rose neurons.

4.2 Integrate-And-Fire Interaction Function

The integrate-and-fire neural model is one of the oldest models of neural behavior. Originally put forth in 1907 by Lapicque [58], and translated in Brunel & van Rossum [59], the integrate-and-fire function reduces the action potential of a neuron to a simple mechanism. In this neural model, the neuron sums up, or integrates, input to the membrane voltage that the neuron is receiving from the dendrites. If the membrane voltage reaches a particular threshold then a delta-like spike occurs. Morena & Short [7] implemented an analogous *Integrate-And-Fire* interaction function that is adapted from the integrate-and-fire neural and laser dynamics seen in Racicot & Longtin [60] and Hall *et al.* [61]. The *Integrate-And-Fire* interaction function is used in this chapter to mediate the communication between neurons due to the similarity with neural firing properties.

Recall from Chapter 3 that a cupolet generates a visitation sequence which corresponds to the symbolic dynamics produced when visiting a sequence of control planes. In Figure 3.8, the control sequence 001 established cupolet C001. As cupolet C001 traverses 1 period, over approximately 76.36 time units, the trajectory visits PS1 twice (consecutively) and visits PS0 once. Thus the visitation sequence is V110. Then, for example, if the first bit of the earliest intersection with cupolet C001 is with PS1 and 9 bits are recorded, we would expect V110110110. This process can be extended for any number of bits. The *Integrate-And-Fire* (IF) interaction function reads in the most recent visitation sequence of bit length Q and returns either a 1 or 0. The return value is a 1 if the bits in the most recent Q length visitation sequence sums to a particular threshold, κ, and a 0 if the threshold κ is not reached. This is abbreviated as IF(Q,κ). As an example, take the 10 bit visitation sequence of C001, V1101101101. Using IF($Q = 3$,$\kappa = 3$), the returned value is 0, however if IF($Q = 4$,$\kappa = 3$) then the returned value is 1. This is because the last $Q = 3$ bits are 101 which does not sum to the $\kappa = 3$ threshold, thus returning a 0. But if Q is increased by 1, then the last $Q = 4$ bits are 1101, which do sum to the $\kappa = 3$ threshold and returns a 1.

Visitation Sequence	Most Recent $Q = 4$ Bits	IF(4,4)	Control Sequence
...0111	0111	0	0
...01111	1111	1	01
...011110	1110	0	010
...0111101	1101	0	0100
...01111011	1011	0	01000
...011110111	0111	0	010000

Table 4.1: Generation of control sequence using IF(4,4) applied to a visitation sequence from C11A.

Visitation Sequence	Most Recent $Q = 4$ Bits	IF(4,3)	Control Sequence
...0111	0111	1	1
...01111	1111	1	11
...011110	1110	1	111
...0111101	1101	1	1111
...01111011	1011	1	11111
...011110111	0111	1	111111

Table 4.2: Generation of control sequence using IF(4,3) applied to a visitation sequence from C11A.

While a cupolet evolves, the most recent Q length visitation sequence is constantly passed to the corresponding function IF(Q,κ) to generate a control bit. In this manner, only the most recent Q bits are considered to establish the control output. One example is given in Table 4.1 where $Q = 4$ and $\kappa = 4$ and another example is given in Table is given in 4.2 where $Q = 4$ and $\kappa = 3$. In both tables, the interaction function is applied to the visitation sequence from the same cupolet C11A visitation sequence V01111011111.

Recall that the control scheme detailed in §3.3 applies a repeated binary control sequence where 1 corresponds to a macrocontrol, or a large kick, and a 0 corresponds to a microcontrol, or a tiny kick. In Table 4.2, the repeated visitation sequence of C11A generates the control sequence 111111 as a result of IF(4,3). Therefore, this interaction function produces only a control bit of 1 that is applied when the trajectory visits a control plane. Additionally, this control sequence is the same control sequence that was used to generate the original cupolet C11A. This implies that if C11A were to interact with a chaotic neuron unidirectionally

via IF(4,3) then this second neuron will generate the cupolet C11 since the applied control sequence is 11 cyclically.

4.2.1 Unidirectional Stabilization

In this section, we consider a unidirectional chain of neurons where communication only moves forward and is mediated by the *Integrate-And-Fire* interaction function. Figure 4.1 shows an example where three neurons are unidirectionally linked. That is, Neuron 1 communicates to Neuron 2, and Neuron 2 communicates to Neuron 3. Initially, Neuron 1 is the only neuron that receives a control sequence, specifically 11. This continues for 2560000 iterations, with $dt = 1/128$, allowing for enough time such that Neuron 1 stabilizes onto cupolet C11A. The first row of Figure 4.1 shows iterations 1280000 to 2560000 where it can be seen that Neuron 1 has stabilized onto cupolet C11A while Neuron 2 and Neuron 3 continue exhibiting chaotic neural firing. Then, between iterations 2560000 and 5120000, Neuron 1's visitation sequence is continually sent into IF(4,3) where the output is a control on Neuron 2. The resultant control, as seen from Table 4.2, is 1, repeatedly. The second row of Figure 4.1 shows iterations 3840000 to 5120000 where it can be seen that Neuron 2 has stabilized onto cupolet C11A via communication with Neuron 1, while Neuron 1 continues to evolve as cupolet C11A. Neuron 3 continues to exhibit chaotic neural firing. From iterations 5120000 to 7680000, Neuron 2 communicates with Neuron 3 via IF(4,3). Row 3 of Figure 4.1 shows iterations 6400000 to 7680000 where it can be seen that Neuron 3 has stabilized onto cupolet, C11B. Neuron 1 and Neuron 2 continue to receive the same controls from the previous iteration segments and thus both continue to evolve as cupolet C11A. The cupolet C11 for Neuron 3 is a different C11 cupolet than Neuron 1 and Neuron 2, reflecting the multiplicity of certain cupolet control sequences that the Hindmarsh-Rose neuron can produce and was previously discussed in Chapter 3. Note that this process is originally triggered through an external control on Neuron 1. If another neuron was added to the chain, or any number of neurons were added, and the interactions between neurons are kept the same,

then this unidirectional stabilization would continue. The next section discusses how mutual stabilization can occur between two bidirectional communicating neurons.

4.3 Bidirectional Model

This section expands on the chaotic stabilization in the previous section by demonstrating how mutual stabilization can explicitly occur between two interacting HR neurons. The mutual stabilization process is analogous to the mutual stabilization reported in Morena & Short [7] and Morena [48] and illustrated previously in Figure 1.1. First, two neurons independently evolve in the chaotic regime. Then, a control sequence is applied to each neuron that is known to generate a cupolet. These two control sequences, which do not have to be identical, are indicative of separate neurons receiving different external signals. Next, continue to apply the control sequence cyclically until each neuron has evolved for a certain number of iterations and stabilized onto a cupolet. Record the visitation sequence of each neuron. This process of generating a single cupolet is identical to the process described in Chapter 3. Once the cupolets are established, the cupolets begin to communicate bidirectionally. That is, Neuron 1 can send information to Neuron 2 and Neuron 2 can send information to Neuron 1 where the method of communication is mediated by the interaction function IF(Q,κ). At this point, the external controls are removed and the IF(Q,κ) interaction function remains the only control on the cupolets. This process continues until both neuron stabilize onto a cupolet. In practice, the simulation time lasts for the same number same number of iterations that it took to establish the original cupolets before communication began, which is enough time to achieve mutual stabilization.

Figure 4.2 shows an example of mutual stabilization for two Hindmarsh-Rose neurons that are communicating via IF(5,4). After an initial chaotic transient, the control sequence 001 is repeatedly applied to Neuron 1 leading to Neuron 1 stabilizing onto cupolet C001. Figure 4.2A shows the resultant cupolet C001 during iterations 1280000 to 2560000. During the same simulation time, control 01 is repeatedly applied to Neuron 2 leading to Neuron 2

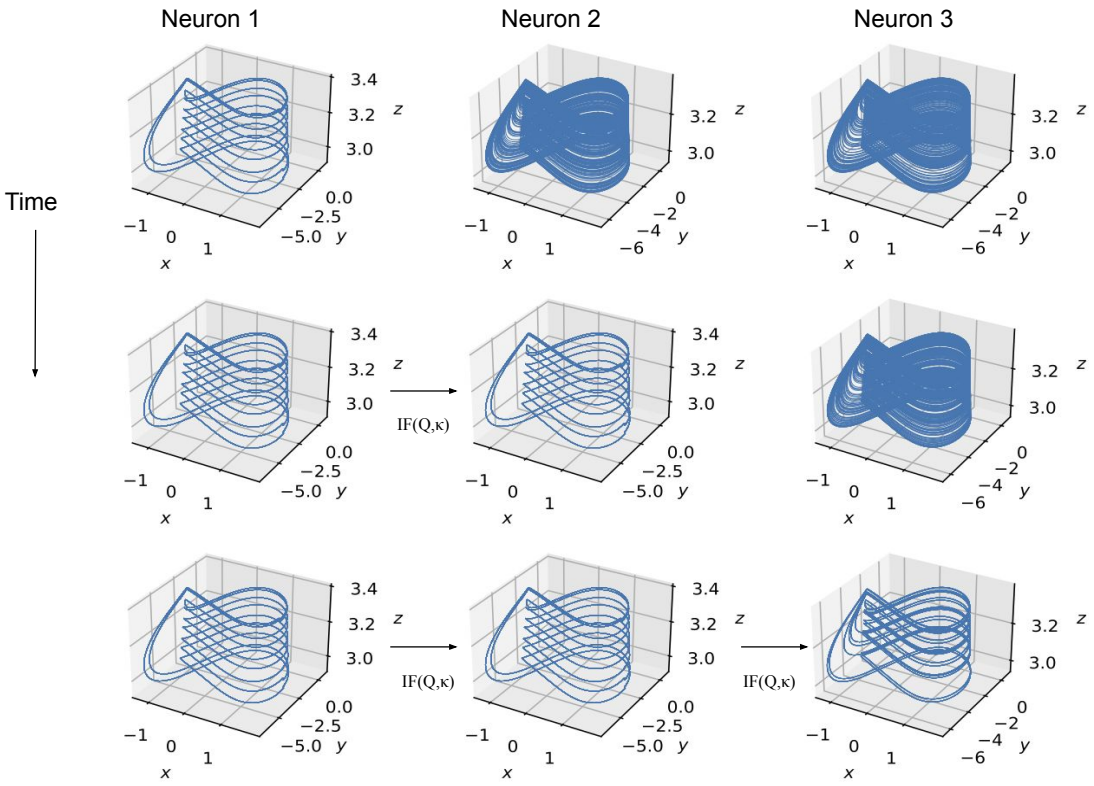

Figure 4.1: Chain of neurons linked through unidirectional coupling. Each column represents an individual neuron, Neuron 1, Neuron 2, and Neuron 3. Each neuron was simulated with a $dt = 1/128$ for $N = 7680000$ iterations after an initial transient starting from a random initial condition. Neuron 1 starts communicating with Neuron 2 via IF(4,3) after iteration 2560000. Neuron 2 starts communicating with Neuron 3 via IF(4,3) after iteration 5120000. Communication continues until simulation time ended. Row 1 plots the iterations 1280000 to 2560000 where only Neuron 1 receives an external control. In row 2, iterations 3840000 to 5120000 are shown and in row 3 iterations 6400000 to 7680000 are shown. Neuron 1 and Neuron 2 form C11A while Neuron 3 forms C11B.

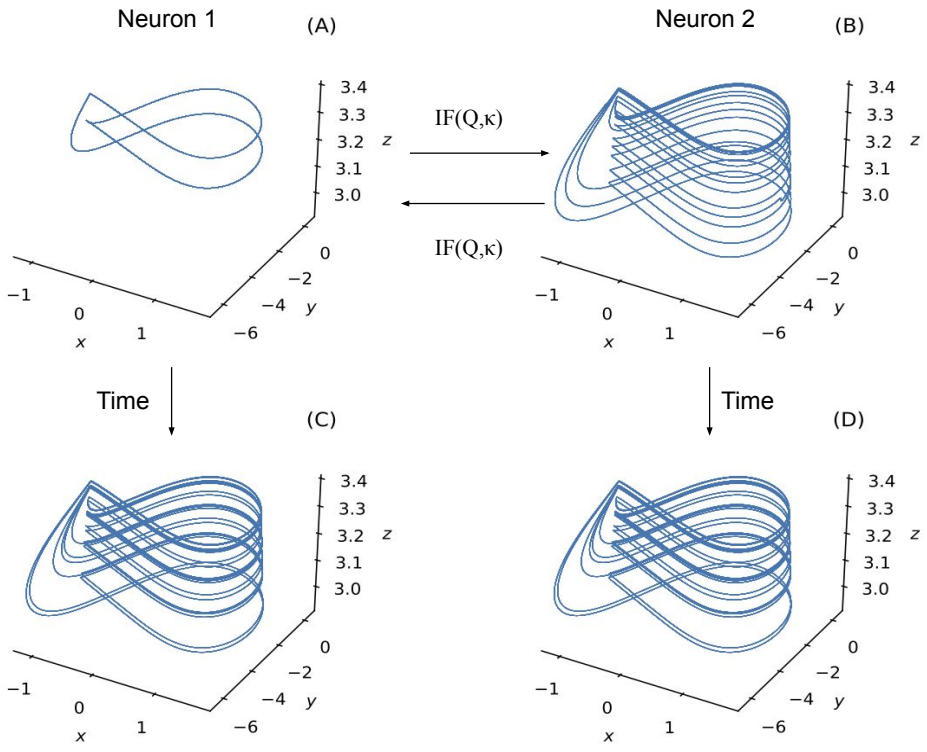

Figure 4.2: Mutual stabilization of two Hindmarsh-Rose neurons. (A) Neuron 1 stabilizes onto cupolet C001 after 2560000 iterations. Iterations 1280000 to 2560000 are plotted. (B) Neuron 2 stabilizes onto cupolet C01 after 2560000 iterations. Iterations 1280000 to 2560000 are plotted. (C) Neuron 1 receives the controls that are determined from the visitation sequence of Neuron 2 via IF(5,4) and replaces the original control sequence 001. This results in the repeated control 11 leading Neuron 1 to stabilize onto cupolet C11B. Iterations 3840000 to 5120000 are plotted. (D) Neuron 2 receives the controls that are determined from the visitation sequence of Neuron 1 via IF(5,4), replacing the original control sequence 01. This results in the repeated control 11 leading Neuron 2 to stabilize onto cupolet C11B. Iterations 3840000 to 5120000 are plotted.

to stabilize onto cupolet C01. Figure 4.2B shows the resultant cupolet C01 during iterations 1280000 to 2560000. Between iterations 2560000 to 5120000, the two neurons begin bidirectional communication that is mediated by IF(5,4). That is, Neuron 1 sends the most recent visitation sequence of length $Q = 5$ to IF(5,4) and the output of IF(5,4) is the next control on Neuron 2. Neuron 2 sends the most recent visitation sequence of length $Q = 5$ to IF(5,4) and the output of IF(5,4) is the next control bit on Neuron 1. For both Neuron 1 and Neuron 2, each visitation sequence of length $Q = 5$ results in an output of 1 from IF(5,4). This results in Neuron 1 stabilizing onto cupolet C11B (Figure 4.2C) and Neuron 2 stabilizing onto cupolet C11B (Figure 4.2D). This is one of the simplest cases of mutual stabilization where heterogeneous cupolets (Neuron 1 and Neuron 2 have stabilized onto different cupolets, C001 and C01, respectively) transitioned to homogeneous cupolets (Neuron 1 and Neuron 2 have stabilized onto the same cupolet, C11B) once the cupolets started interacting.

The mutual stabilization produced in Figure 4.2 is dependent on the specific interaction function between the two neurons. If the interaction function is IF(4,4), then the last 4 bits of the respective visitation sequence are used instead of 5 bits. An IF(4,4) output value of 1 corresponds to 4 input bit values that are all 1 which means that 4 consecutive spikes occurred. Therefore, the only time the IF(4,4) function results in 1 is when the symbolic dynamics of a cupolet has a burst of at least 4 spikes. Figure 4.3 shows the same initial cupolets as Figure 4.2 yet no mutual stabilization occurs because of the difference in interaction function. An IF(4,4) leads to mainly 0 control bits (microcontrols) rather than 1 control bits (macrocontrols) and as a result the neurons do not mutually stabilize.

Figure 4.3A shows the cupolet C001 during iterations 1280000 to 2560000. During the same simulation time, the external control 01 is repeatedly applied to Neuron 2 leading Neuron 2 to stabilize onto cupolet C01. Figure 4.3B shows the cupolet C01 during iterations 1280000 to 2560000. Between iterations 2560000 to 5120000, the bidirectional communication between then neurons is mediated via IF(4,4). That is, Neuron 1 sends the most recent visitation sequence of length $Q = 4$ to IF(4,4) and the result becomes the next control on

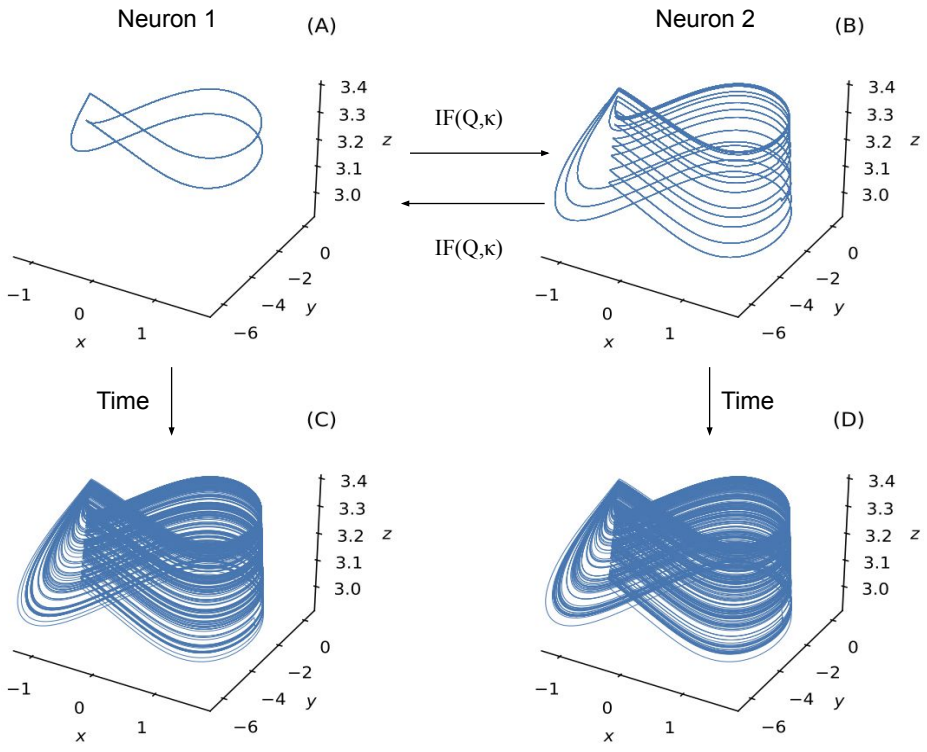

Figure 4.3: Failed mutual stabilization of two communicating Hindmarsh-Rose neurons. (A) Neuron 1 stabilizes onto cupolet C001 after 2560000 iterations. Iterations 1280000 to 2560000 are plotted. (B) Neuron 2 stabilizes onto cupolet C01 after 2560000 iterations. Iterations 1280000 to 2560000 are plotted. (C) Neuron 1 receives the controls that are determined from the visitation sequence of Neuron 2 via IF(4,4) and replaces the original control sequence 001. The resultant controls are not sufficient to establish a cupolet. Iterations 3840000 to 5120000 are plotted. (D) Neuron 2 receives the controls that are determined from the visitation sequence of Neuron 1 via IF(4,4), and replaces the original control sequence 01. The resultant control is not sufficient to establish a cupolet. Iterations 3840000 to 5120000 are plotted.

Neuron 2. Neuron 2 sends the most recent visitation sequence of bit length $Q =$ to IF(4,4) and the result becomes the next control on Neuron 1. The visitation sequences produced by Neuron 1 and Neuron 2 result in an output of mainly microcontrols, 0, from IF(4,4). This does not result in Neuron 1 (Figure 4.3C) or Neuron 2 stabilizing onto a cupolet (Figure 4.2D) during the simulation time, even though the initial cupolet formed is identical to the scenario in Figure 4.2. Both neurons return to chaotic neural firing since the controls did not drive the system to stabilize the chaotic trajectories onto cupolets.

4.4 Discussion

The first two chapters of this thesis, Chapter 1 and Chapter 2, focused on demonstrating how mutual stabilization can occur between two FitzHugh-Nagumo neurons. This result was motivated by mutual stabilization of communicating cupolets of the double scroll oscillator, but the stabilization was achieved through neural learning that changed the nature of the synaptic strength and effectively led the system out of a chaotic regime. that occurred when the communication was mediated by the *Integrate-And-Fire* interaction function. The mutual stabilization that resulted from the interacting FHN neurons allowed for a proof-of-concept for mutual stabilization in neural systems. In Chapter 3 we demonstrated how cupolets can be generated by following an approach equivalent to that used the cupolets generated from the double scroll oscillator, where the *Integrate-And-Fire* interaction function was employed. This chapter focused on connecting the three previous chapters by demonstrating how chaotic stabilization can occur with communicating chaotic neurons, where the neurons are always in a chaotic region of parameter space.

The communication between neurons in this chapter was mediated by the *Integrate-And-Fire* interaction function from Morena & Short [7]. This interaction function, abbreviated as IF(Q,κ), receives the symbolic dynamics of a signal (length Q) and returns a control bit of 1 if the symbolic dynamics sum to κ or 0 if not. Biologically, neurons receive input, commonly called a stimulus, and if the stimulus changes the membrane potential such that

the membrane potential reaches a certain threshold, an action potential is released. The action potential is the same as the spike that occurs in the dynamical model of a neuron. Similarly, the interaction function IF(Q,κ) can be considered as an abstraction of a stimulus that a neuron receives. If the receiving neuron is correctly stimulated then the neuron's dynamics shift towards a specific periodic orbit and that periodic orbit is persistent as long as the communication continues. This is evident from Figure 4.2. However, if the control, acting as the stimulus, is not appropriate then the receiving neuron will continue to evolve as if no stimulus is applied, as can be seen in Figure 4.3. This implies that through this communication scheme the interaction between neurons is crucial to eliciting certain persistent, periodic behaviors, but not the parameter set of the neuron.

Recall Figure 4.1 where three neurons are linked unidirectionally. At a particular time, Neuron 1 is stabilized onto a particular cupolet. Information about that cupolet is communicated to Neuron 2 through IF(4,3) and acts as a stimulus on Neuron 2. In the example discussed in this chapter, this locks Neuron 2 onto the same periodic orbit as Neuron 1. The communication continues to a third neuron, however, the resultant dynamics result in Neuron 3 stabilizing onto a different type of cupolet (C11B instead of C11A). As discussed in Chapter 3, homologous cupolets exist because the control sequence happened to be applied on different initial bins for Neuron 3 than Neuron 1 or Neuron 2. This different response is akin to a signal propagating down a chain of identical neurons in which some are driven into identical periodic states (Neuron 1 and Neuron 2) but results in a different periodic state when a neuron in a different part of phase space is encountered. This process can occur for any number of unidirectional neurons, assuming the interactions produce the controls that can generate cupolets. This process is similar to when a signal felt by one neuron propagates information to another neuron, thus forming a neural chain called a nerve. The communication established by IF(Q,κ) mediates the communication between the neurons and an example of a signal propagating down a nerve is given as Figure 4.1.

The mutual stabilization presented in this chapter was produced following Morena &

Short [7] but with chaotic neurons instead of the double scroll oscillator. Dependent upon the communication between the neurons, mutual stabilization may occur or the neurons may continue to evolve in a chaotic regime. This suggests a possible role for the existence of chaos in neurons. There is evidence that neurons and neural networks exhibit chaotic behavior. In particular, Rabinovich & Abarbanel [42] cites central pattern generators, specifically lobster central pattern generators, as a prime example of the role of chaos in neurons. We suggest that the mutual stabilization presented in this chapter is the simplest case of a model of a central pattern generator. A similar example of a role for chaos in neurons was suggested in Chapter 2 where chaotic stabilization through interaction between two FitzHugh-Nagumo models occurred. This example suggested that the stabilization or locking of neurons into a specific state may be related to a form of "stored memory." That is, when the system receives a specific stimulus, which leads to mutual stabilization, it retains information about the original signal that is reflected by the specific persistent, periodic orbit.

In the following chapter, we expand the results of this chapter to larger neural networks. First, we address the case of mutual stabilization existing at the end of a nerve. A control sequence stabilizes a chaotic neuron onto a cupolet and information of the cupolet propagates down the chain via the interaction function IF(Q,κ) until terminating at the final two neurons. The final two neurons could potentially mutually stabilize if the right communication signal is produced, as mediated via an interaction function. A second interesting case we address is feedback stabilization where a larger chain of unidirectional neurons is considered but the last neuron in the chain communicates back with the first neuron. The feedback between the last neuron and the first neuron can lock the entire loop of neurons into a stabilized state.

CHAPTER 5
NETWORK CHAOTIC STABILIZATION

5.1 Motivation

In Chapter 2 and Chapter 4, we reported how different models of interacting neurons can mutually stabilize. In Chapter 2, it was discussed how a system of two FitzHugh-Nagumo neurons can exhibit chaotic firing when coupled together. Then, we demonstrated how the FHN system could transition to persistent, periodic behavior through a sigmoidal, synaptic learning rule activated at the presence of an external signal. In Chapter 3, we showed how to stabilize a chaotic Hindmarsh-Rose neuron onto cupolets through the use of binary control sequences that implemented microcontrols and macrocontrols. The results of Chapter 2, Chapter 3, and Morena & Short [7] led to the results in Chapter 4 where a signal propagated down a nerve (unidirectional stabilization) and where bidirectional communicating neurons could mutually stabilize. In the Chapter 4 scenarios, communication between the neurons was mediated by the *Integrate-And-Fire* interaction function. In this chapter, we expand the results of Chapter 4 to simple networks of four neurons. Specifically, the focus is on two different network configurations where each configuration can be considered as a building block for more complex and larger neural networks that are typically found in biological organisms.

5.1.1 Networks

The first network demonstrates another case of mutual stabilization. This case occurs at the end of a chain of neurons, representing a nerve, and it involves both the unidirectional

and bidirectional interactions that were discussed in Chapter 4. The chain of neurons will be comprised of four neurons, where each neuron is initially exhibiting chaotic neural firing. At the beginning of the chain, Neuron 1 receives a binary control signal and this leads to Neuron 1 to stabilize onto a cupolet. Information about this cupolet is passed through IF(Q,κ) to the next neuron, Neuron 2, which stabilizes onto a cupolet as a result of the interaction between the neurons. This process repeats down the chain until the last neuron, Neuron 4, stabilizes onto a cupolet. That is, Neuron 1 communicates to Neuron 2, Neuron 2 communicates to Neuron 3, and then Neuron 3 communicates to Neuron 4. Then the last neuron, Neuron 4, communicates back to Neuron 3 through IF(Q,κ) and the output of IF(Q,κ) replaces the communication that Neuron 3 receives from Neuron 2 (the represents the passing of the impulse that traveled up the nerve). The communication from Neuron 3 to Neuron 4 is unchanged. At this point, Neuron 3 talks to Neuron 4 and Neuron 4 talks to Neuron 3, but all other communication ceases between other neurons in the chain. The end result is that Neuron 3 and Neuron 4 can become into mutual stabilization while Neuron 1 and Neuron 2 no longer receive any input and thus return to chaotic neural firing. A diagram of this network is given in Figure 5.1. Each subfigure in Figure 5.1 contains four circles that represent each individual neuron in the chain. If the circles are green then that neuron has stabilized onto a cupolet and if the circles are yellow then the neuron is exhibiting chaotic neural firing. In Figure 5.1A, Neuron 1 has stabilized onto a cupolet from an external binary control sequence, that represents an external stimulus, while the other three neurons are exhibiting chaotic neural firing. Then, in Figure 5.1B, Neuron 1 communicates to Neuron 2 via IF(Q,κ) which leads to Neuron 2 stabilizing onto a cupolet. This process repeats in Figure 5.1C where Neuron 1, Neuron 2, and Neuron 3 have all stabilized onto cupolets because of the communication that is mediated by the interaction function with the preceding neuron. Then, in Figure 5.1D, mutual stabilization occurs when Neuron 3 and Neuron 4 bidirectionally communicate via IF(Q,κ) while Neuron 1 and Neuron 2 no longer receive any external stimulus or stimulus from any of the neurons. This leads Neuron 3 and Neuron 4 to

mutually stabilize and lock into persistent, periodic neural firing while Neuron 1 and Neuron 2 return to chaotic neural firing. This scenario is called chain mutual stabilization and an example is presented in §5.2.1.

The second network discussed is a loop of four neurons. Similar to the first case, a signal propagates down a chain of four neurons and the neural communication is mediated by IF(Q,κ) that subsequently stabilizes each respective neuron onto a cupolet in a manner similar to that described in Figure 5.1. After Neuron 4 stabilizes onto a cupolet, Neuron 4 communicates back to Neuron 1 where the communication is mediated by IF(Q,κ). The output of IF(Q,κ) replaces the binary external control sequence that originally stabilized Neuron 1 onto a cupolet. All other communication between the neurons remains the same. This can result in each neuron in the loop stabilizing onto a cupolet and we call this feedback stabilization. This is similar to the mutual stabilization case from Chapter 4 except the entire network locks into persistent, periodic neural firing instead of just two neurons that are bidirectionally communicating. A diagram of this case is provided in Figure 5.2. Figures 5.2A-C are exactly the same as the corresponding Figures 5.1A-C. However, in Figure 5.2D, Neuron 4 has stabilized onto a cupolet and begins communicating with Neuron 1, that is again mediated by IF(Q,κ). This communication that Neuron 1 receives replaces the original external binary control sequence on Neuron 1. However, if any one of the neurons failed to receive the controlling stimulus, then that neuron would revert back to chaotic neural firing since that is the default state of each neuron.

This chapter concludes by summarizing the results of each case and discussing the biological impact of the results.

5.2 Results

5.2.1 Chain Mutual Stabilization

In this section we report the results of a signal propagating down a chain of neurons, representative of a nerve, where the final four neurons in the nerve are analyzed. The goal of

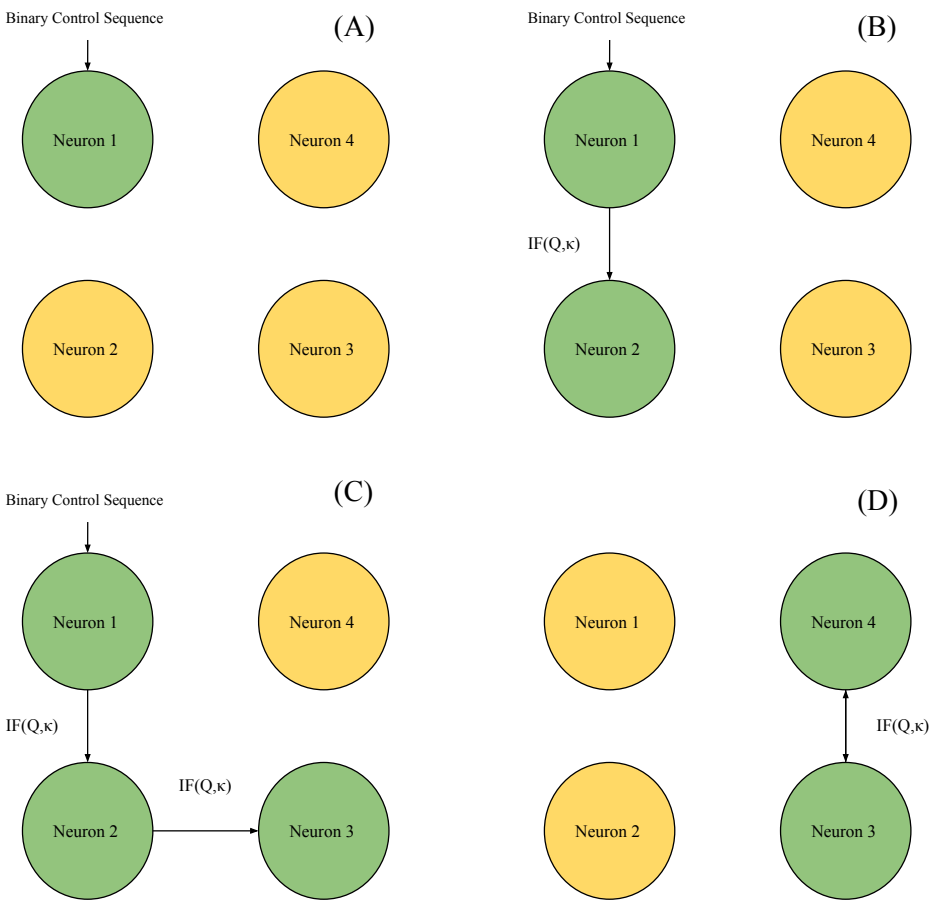

Figure 5.1: Schematic of a chain of neurons that results in mutual stabilization in the final two neurons. A yellow circle means the neuron is exhibiting chaotic neural firing and a green circle means the neuron has stabilized onto a cupolet. (A) Neuron 1 receives a binary control sequence that stabilizes Neuron 1 onto a cupolet. Neurons 2, 3, and 4 all exhibit chaotic neural firing. (B) Neuron 1 communicates with Neuron 2 and the communication is mediated by IF(Q,κ). This results in Neuron 2 stabilizing onto a cupolet. (C) Neuron 2 communicates with Neuron 3 and the communication is mediated by IF(Q,κ). This results in Neuron 3 stabilizing onto a cupolet. (D) Neuron 3 communicates to Neuron 4 and the communication is mediated by IF(Q,κ). Neuron 4 communicates back to Neuron 3. The communication from Neuron 4 to Neuron 3 is mediated by IF(Q,κ) and replaces the communication Neuron 3 received from Neuron 2. If the communication results in a control sequence capable of stabilizing Neuron 3 onto a cupolet, then Neuron 3 and Neuron 4 will mutually stabilize as long as the interaction persists. Neurons 1 and 2 will return to chaotic neural firing since no input is applied to either neuron and the chaotic behavior is the default state of each neuron.

Figure 5.2: Schematic of feedback stabilization involving four neurons. A yellow circle means the neuron is exhibiting chaotic neural firing and a green circle means the neuron has stabilized onto a cupolet. (A) Neuron 1 receives a binary control sequence that stabilizes Neuron 1 onto a cupolet. Neurons 2, 3, and 4 all exhibit chaotic neural firing. (B) Neuron 1 communicates with Neuron 2 and the communication is mediated by IF(Q,κ). This results in Neuron 2 stabilizing onto a cupolet. (C) Neuron 2 communicates with Neuron 3 and the communication is mediated by IF(Q,κ). This results in Neuron 3 stabilizing onto a cupolet. (D) Neuron 3 communicates to Neuron 4 and the communication is mediated by IF(Q,κ). Neuron 4 stabilizes onto a cupolet and begins feedback to Neuron 1, that is mediated again by IF(Q,κ). The feedback replaces the binary sequence that Neuron 1 originally received. As a result, each neuron stabilizes onto a cupolet and feedback stabilization now exists. Without the interactions between the neurons, each neuron would return to chaotic dynamics since that is the default state of each neuron.

the network is to propagate the signal to the neural structure at the end of the chain that is represented by Neuron 3 and Neuron 4. The neurons only have unidirectional communication that is mediated by IF(Q,κ) until each neuron has stabilized onto a cupolet. Then, the last two neurons have bidirectional communication that results in mutual stabilization, as illustrated in Figure 5.1 and the first two neurons no longer receive inputs and return chaotic neural firing. Figure 5.3 shows the resulting phase portrait of each neuron for a specific example of chain mutual stabilization. The rows of Figure 5.3 are snapshots in time and the columns of Figure 5.3 represent each neuron. While moving down the figure, each row contains new interactions within the network. In row 1, Neuron 1 receives an external binary control input that represents an external stimulus. Neuron 1 receives the control sequence 11 (repeated cyclically) that results in Neuron 1 stabilizing onto cupolet C11A (top row, first column). Neuron 1 then begins communication with Neuron 2 via IF(Q,κ) where $Q = 4$ and $\kappa = 3$. This also results in controls of 11 (repeated cyclically) for Neuron 2. Neuron 3 and Neuron 4 continue exhibiting chaotic dynamics and have not yet received any external input. Neuron 2 stabilizes onto a cupolet, C11A, (second row, second column) and begins communication with Neuron 3 via IF(5,3) in row 3. The process continues and Neuron 3 stabilizes onto a cupolet, C11A (third row, third column) and starts to communicate with Neuron 4 via IF(5,3). In row 4, the signal continues to propagate and Neuron 4 stabilizes onto a cupolet, C11B (fourth row, fourth column). This results in Neuron 4 stabilizing onto a cupolet and all other neurons have continued the cupolet dynamics. In the last row, row 5, the only interactions that occur are Neuron 3 communicating with Neuron 4 and Neuron 4 communicating back to Neuron 3. The input to Neuron 1 and the inter-neuron communication ends between Neuron 1 to Neuron 2 and Neuron 2 to Neuron 3. Due to the bidirectional communication of Neurons 3 and Neuron 4, Neurons 3 and Neuron 4 lock into persistent, periodic neural firing and therefore these neurons have mutually stabilized. This can be seen in the last row where Neuron 3 continues to evolve as cupolet C11A and Neuron 4 continues to evolve as cupolet C11B, showing homologous cupolets in mutual stabilization.

Both Neuron 1 and Neuron 2 revert back to chaotic neural firing because there is no input and the chaotic behavior is the default state of each neuron. The simulation lasts for 100000 time units, or 12800000 iterations. The first row is the snapshot of the simulation from time units 10000 to 20000. Each subsequent row displays the corresponding snapshot 20000 time units later from the previous row. That is, row 2 displays time units 30000 to 40000, row 3 displays time units 50000 to 60000 and so on. This length of time is chosen so that each cupolet can be clearly seen and allows the same simulation time for the formation of the cupolets as was discussed in the previous two chapters.

Figure 5.4 displays the corresponding spike raster plots of each image in Figure 5.3. Each vertical line is the exact time of the peak of a spike of the corresponding neuron. These plots allow for the bursting pattern to be clearly seen as well as whether or not the neurons are spiking at the same time. In the first spike raster plot, Neuron 1 exhibits the firing pattern of cupolet C11A while Neuron 2, Neuron 3, and Neuron 4 continue to exhibit chaotic neural firing. In the second spike raster plot, the timing of the spikes from Neuron 1 are just ahead of the spikes from Neuron 2 even though both neurons have stabilized onto the same cupolets, C11A. Neuron 3 and Neuron 4 continue to exhibit chaotic neural firing. The third spike raster plot shows the spike timing of Neuron 3 stabilized onto cupolet C11A while only Neuron 4 is exhibiting chaotic neural firing. In the spike fourth raster plot, Neuron 1, Neuron 2, and Neuron 3 continue firing as cupolet C11A while Neuron 4 exhibits cupolet C11B firing. Every neuron has stabilized onto a cupolet. Neuron 1, Neuron 2, and Neuron 3 are all out of phase with each other. In the final spike raster plot, corresponding to the final row in Figure 5.3, Neuron 1 and Neuron 2 no longer receive any input and return to chaotic neural firing. Neuron 3 and Neuron 4 have mutually stabilized with Neuron 3 having stabilized onto cupolet C11A and Neuron 4 having stabilized onto cupolet C11B. As long as this interaction continues, Neuron 3 and Neuron 4 will continue persistent, periodic firing while Neuron 1 and Neuron 2 will revert to chaotic neural firing.

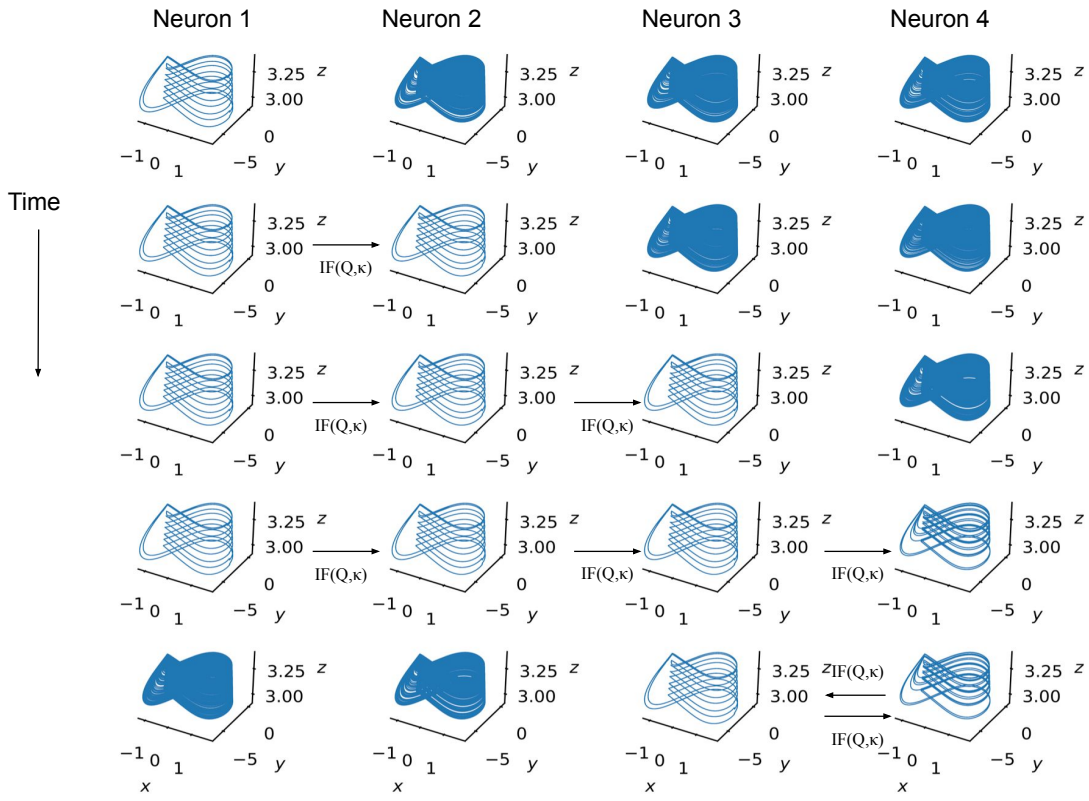

Figure 5.3: Phase plots of four neurons in a chain that results in mutual stabilization between the last two neurons and is based on the diagram in Figure 5.1. Each column represents a neuron while the rows are snapshots in time. In each row, the impact of the interactions moves to the right. In the first row, Neuron 1 has stabilized onto cupolet C11A and the other three neurons exhibit chaotic neural firing. The second row has Neuron 1 continuing as cupolet C11A, but communication that is mediated via IF($Q = 5$,$\kappa = 3$) has started with Neuron 2. Neuron 2 stabilizes onto cupolet C11A. In the third row, Neuron 1 and Neuron 2 continue the same interactions that resulted in cupolet C11A in both neurons. Neuron 2 communicates via IF($Q = 5$,$\kappa = 3$) to Neuron 3, which results in Neuron 3 stabilizing onto cupolet C11A. In the fourth row, Neuron 3 communicates to Neuron 4 via IF($Q = 5$,$\kappa = 3$), which results in Neuron 4 stabilizing onto cupolet C11B. In the final row, Neuron 3 and Neuron 4 mutually stabilize because of the bidirectional communication via IF($Q = 5$,$\kappa = 3$). Neuron 1 and Neuron 2 no longer receive any input and return to chaotic neural firing since that is the default state of each neuron.

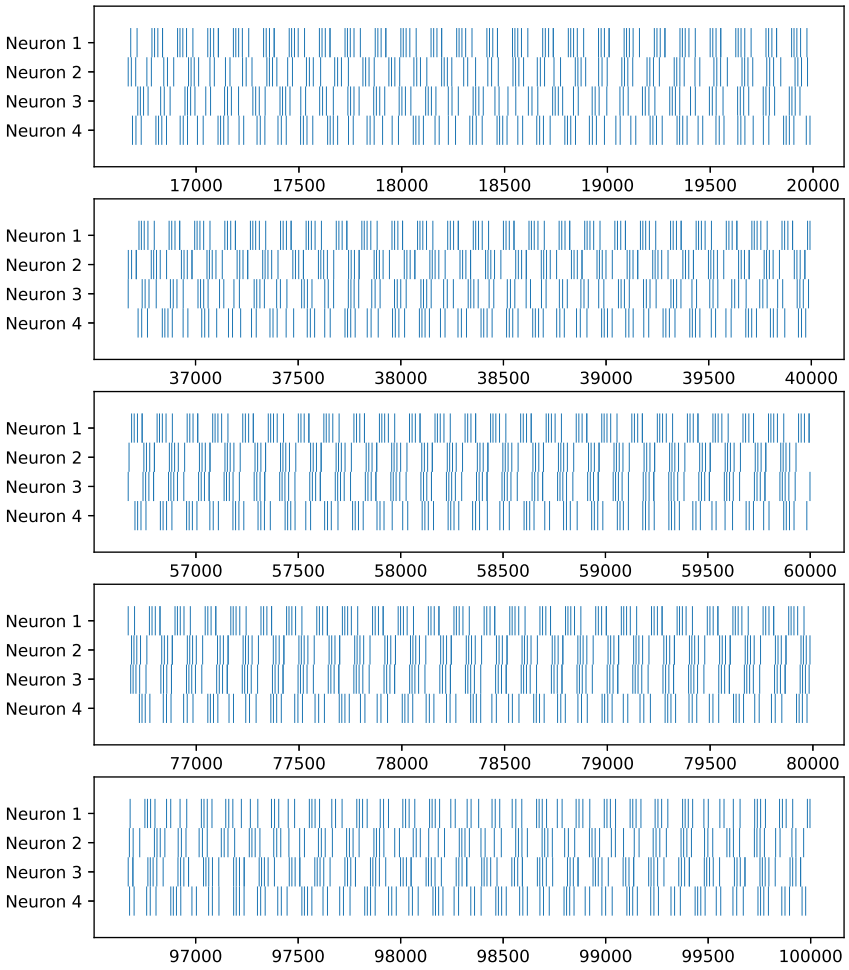

Figure 5.4: Spike raster plots of Figure 5.3, where each image is a raster plot of the corresponding image in Figure 5.3. In the first spike raster plot (first row), Neuron 1 exhibits the spiking pattern of cupolet C11A and Neuron 2, Neuron 3, and Neuron 4 exhibit chaotic neural firing. In the second spike raster plot (row 2), Neuron 1 communicates to Neuron 2 leading to both neurons exhibiting the spiking pattern of cupolet C11A. Neuron 1 out of phase with Neuron 2. Neuron 3 and Neuron 4 continue with chaotic neural firing. In the third spike raster plot (row 3), Neuron 3 exhibit the spiking pattern of cupolet C11A. Neuron 1 and Neuron 2 continue the same firing pattern (cupolet C11A) and Neuron 4 exhibits chaotic neural. In the spike fourth raster plot (row 4), Neuron 1, Neuron 2, and Neuron 3 exhibit cupolet C11A firing while Neuron 4 exhibits cupolet C11B firing. The final raster plot (row 5) corresponds to the bidirectional interaction between Neuron 3 and Neuron 4, resulting in homologous cupolet mutual stabilization. Neuron 1 and Neuron 2 have returned to chaotic neural firing.

5.2.2 Feedback Stabilization

In this section, we discuss a specific instance of the feedback stabilization network configuration described in Figure5.2. Similar to Figure 5.3, the rows of Figure 5.5 are snapshots in time and the columns of Figure 5.5 represent each neuron. In each row a new interaction has occurred. In row 1, Neuron 1 receives an external binary control input which results in Neuron 1 stabilizing onto a cupolet, C11A, while the other neurons continue to exhibit chaotic firing. In row 2, Neuron 1 communicates with Neuron 2 via IF($Q = 5$,$\kappa = 3$) and Neuron 2 stabilizes onto cupolet C11A. In row 3, Neuron 1 continues communication with Neuron 2 via IF(5,3) and Neuron 2 begins communication with Neuron 3 via IF(5,3), which results in Neuron 3 stabilizing onto cupolet C11B. Then, in row 4, Neuron 4 receives input from Neuron 3 through IF(5,3) and Neuron 4 stabilizes onto C11B. In the last row, row 5, Neuron 4 feeds back to Neuron 1 via IF(5,3) by replacing the external binary sequence with the output of IF(5,3). Neuron 1 remains stabilized onto cupolet C11A with the feedback from Neuron 4. All other neural communication remains the same. The network has now achieved feedback stabilization where every neuron is locked into persistent, periodic dynamics that will continue as long as the interaction persists. The simulation lasts for 100000 time units, or 12800000 iterations. The first row is the snapshot of the simulation from time units 10000 to 20000. Each subsequent row displays the corresponding snapshot 20000 time units later from the previous row, so row 2 displays time units 30000 to 40000, row 3 displays time units 50000 to 60000 and so on. This length of time is chosen so that each cupolet can be clearly seen and gives the same simulation time for the formation of the cupolets that was discussed in the previous two chapters.

Figure 5.6 shows the corresponding spike raster plots of each row in Figure 5.5. In the first spike raster plot (row 1), Neuron 1 exhibits the spiking pattern of cupolet C11A while Neuron 2, Neuron 3, and Neuron 4 continue to exhibit chaotic neural firing. In the second spike raster plot (row 2), Neuron 1 continues to exhibit the firing pattern of cupolet C11A and is out of phase with the cupolet C11A spiking pattern of Neuron 2. Neuron 3 and

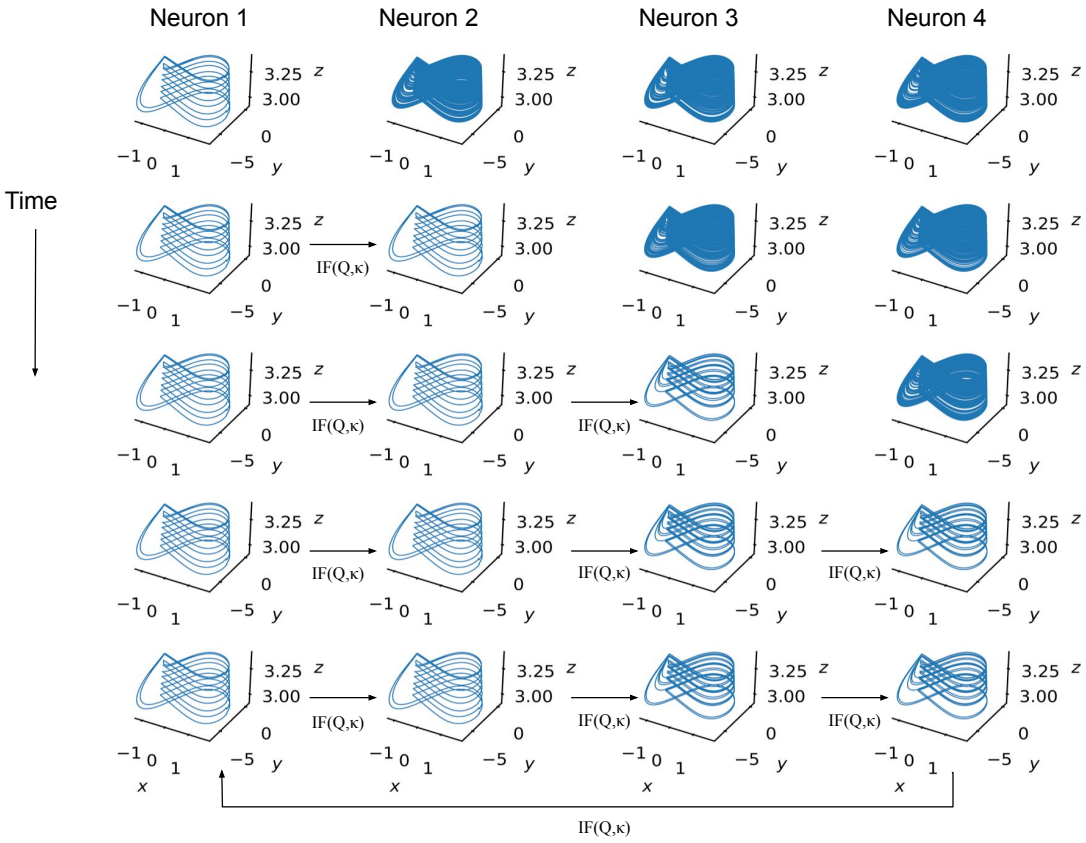

Figure 5.5: Phase plots of four neurons that result in feedback stabilization and as illustrated in Figure 5.2. Each column represents a neuron while the rows are phase space plots corresponding to long snapshots in time. In the first row, Neuron 1 has stabilized onto cupolet C11A and the other three neurons exhibit uncontrolled, chaotic neural firing. The second row shows Neuron 1 stabilized onto C11A but communication begins to Neuron 2 via IF(5,3), which results in Neuron 2 stabilizing onto cupolet C11A. In the third row, Neuron 1 and Neuron 2 continue the same interactions and Neuron 2 begins communication with Neuron 3 via IF(5,3) which results in Neuron 3 stabilizing onto cupolet C11B. In the fourth row, Neuron 3 communicates to Neuron 4 via IF(5,3) which results in Neuron 4 stabilizing onto cupolet C11B. In the final row, Neuron 4 communicates back to Neuron 1 through IF(5,3) and Neuron 1 remains stabilized onto cupolet C11A. The network locks into feedback stabilization.

Neuron 4 continue to exhibit chaotic neural firing. The third spike raster plot (row 3) now shows the spiking pattern of cupolet C11B for Neuron 3 and only Neuron 4 is still exhibiting chaotic neural firing. Neuron 1 and Neuron 2 continue cupolet C11A firing patterns. In the fourth raster plot, Neuron 1 and Neuron 2 have cupolet C11A firing patterns while Neuron 3 and Neuron 4 exhibit cupolet C11B firing patterns. Every neuron has stabilized onto a cupolet. Neuron 1 is out of phase with cupolet C11A for Neuron 2. Similarly, Neuron 3 is out of phase with cupolet C11B for Neuron 4. In the final raster plot (row 5), corresponding to the final row in Figure 5.5, Neuron 4 feeds back into Neuron 1 but Neuron 1 remains stabilized onto cupolet C11A. Neuron 3 remains out of phase with cupolet C11B for Neuron 4. As long as this interaction continues, all four neurons will be locked into this feedback stabilization due to the interactions within the network.

5.2.3 Numerics

Each network was simulated using the control scheme, cupolets, and interaction function discussed in the previous chapters. The time step for the numerical simulation was $dt = 1/128$ and the integration scheme was the explicit Runge-Kutta 4th order method. The total simulation time was $t_f = 10^5$, resulting in 12,800,000 iterations. Each new interaction was allowed 2560000 total iterations ($t = 20000$) before a new interaction occurred. Each row in Figure 5.3 and Figure 5.5 corresponds to the final half, or final 1280000 iterations as discussed in each corresponding section.

5.3 Conclusion

In this chapter, we extended the results of chaotic stabilization of a unidirectional chain of neurons and mutual stabilization that was previously reported in Chapter 4. The chain mutual stabilization network, seen in §5.2.1, combined the results of the unidirectional chain and mutual stabilization seen in Chapter 4 to demonstrate how mutual stabilization can occur from a signal coming down a chain of neurons which represents a nerve. The signal

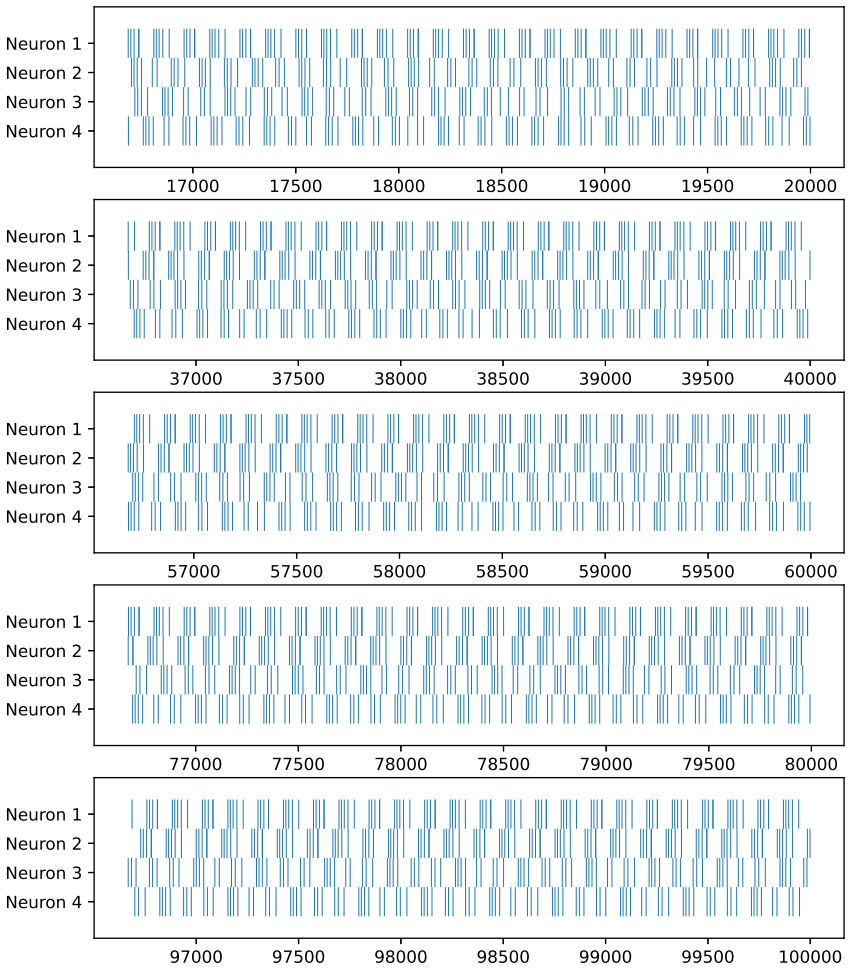

Figure 5.6: In the first spike raster plot (row 1), Neuron 1 exhibits the spiking pattern of cupolet C11A while Neuron 2, Neuron 3, and Neuron 4 exhibit chaotic neural firing. In the second spike raster plot (row 2), Neuron 1 communicates to Neuron 2 leading to both neurons exhibiting cupolet C11A firing patterns. Neuron 1 is out of phase with Neuron 2. Neuron 3 and Neuron 4 continue with chaotic neural firing. In the third spike raster plot (row 3), Neuron 3 has formed C11B due to communication with Neuron 2. Neuron 1 and Neuron 2 continue the firing pattern of cupolet C11A and Neuron 4 exhibits chaotic neural firing. In the fourth spike raster plot, Neuron 1 and Neuron 2 exhibit the firing pattern of cupolet C11A while Neuron 3 and Neuron 4 exhibit the firing pattern of cupolet C11B. The final spike raster plot (row 5) corresponds to the same interaction as the fourth spike raster plot except that Neuron 4 replaces the input on Neuron 1. The network has locked into feedback stabilization but the firing patterns are out of phase.

propagates along the neural pathway until reaching a neural structure that is represented here by either the two final neurons in this network (Neuron 3 and Neuron 4). This structure then retains information about the signal by locking into a mutually stabilized state that persists as long as the interaction between the neurons continue. The neurons further down the chain no longer need to pass information since it is assumed that the stimulus has passed through Neuron 1. These neurons (Neuron 1 and Neuron 2) stop all interaction within the system and return to chaotic neural firing, ready to receive another signal.

The feedback stabilization network in §5.2.2 represents a larger structure that locks into a stabilized state. The last neuron in the chain feeds back to the first neuron and replaces the original stimulus on the first neuron that triggered the chain of stabilization. The entire network locks into a stabilized persistent, periodic state that would not exist without the feedback interaction between the neurons. There is only unidirectional communication between any pair of neurons, yet each neuron becomes stabilized because of the feedback from Neuron 4 to Neuron 1. Neuron 1 and Neuron 2 stabilized onto cupolet C11A while Neuron 3 and Neuron 4 stabilized onto cupolet C11B. The network locks into feedback stabilization yet is a phase difference in the firing between any pair of neurons. These results can be related to the results from Rabinovich & Abarbanel [42] about central pattern generators.

The interaction between the neurons in each of the networks in this chapter was always mediated via IF(5,3). The initial cupolet on Neuron 1 was the same in each network, cupolet C11A. By keeping the same initial cupolet and the same interaction function between neurons, we are able to show that the resultant stable dynamics of the network was dependent on the neural connections within the network. In the absence of these connections, each neuron would have no input and would exhibit chaotic dynamics since that is the default state of each neuron. Both networks discussed in this chapter are among the simplest types of networks that can exhibit a form of chaotic stabilization. This provides a foundation for building larger neural networks to explore further more complicated network dynamics.

Each of the interactions presented in Chapter 3, Chapter 4, and this chapter focused on

instantaneous kicks and binary control sequences following the results in Morena & Short [7]. In the next chapter, we briefly report the initial results of an algorithm to reproduce cupolets using a continuous pulse instead of instantaneous kicks. The goal of this new method is to replace the instantaneous interactions between the neurons with smooth pulses that are more closely related to observed neural stimuli and can also elicit the same results seen in these chapters.

In future work that is beyond the scope of this thesis, we plan to investigate how different values of Q and κ impact the stabilization seen in each network. Additionally, we will examine how the network responds when the connections between neurons do not all reflect the same interaction function IF(Q,κ). This future research will explore how homogeneous connections (reported in this chapter) are different from heterogeneous connections where different synapses are governed by different interaction functions.

CHAPTER 6
SMOOTH CUPOLET TARGETING ALGORITHM: SCUTA

6.1 Motivation

In the previous three chapters, we showed examples of how controls applied to chaotic Hindmarsh-Rose neurons led the neurons to stabilize onto cupolets. These controls were instantaneous perturbations applied to the trajectory upon intersection with each control plane. When the trajectory intersects with a control plane and the control bit is a 0, only a tiny perturbation called a microcontrol is applied. Recall that when the control bit is a 1, a larger, predetermined perturbation called a macrocontrol is applied along the control plane. The binary control sequence that leads to the stabilization of the neuron onto a cupolet can be considered as the overall outcome of the inputs a biological neuron receives, where the kicks are an abstracted way to represent the impact of the impulses received from nearby neurons from signals coming up a chain of neurons. However, mathematical models of neurons can also receive a continuous signal as an input (similar to Chapter 1 and Chapter 2). This chapter briefly describes the initial results of an algorithm that transforms these instantaneous perturbations into a smooth signal. In the next section we detail the approach to the algorithm and in the section afterwards we provide a few examples of the algorithm.

6.2 SCUTA

The goal of the algorithm is to spread the instantaneous perturbation of a desired control over many steps such that the same perturbation is achieved. The algorithm is called

Smooth Cupolet Targeting Algorithm and is abbreviated as SCUTA. SCUTA is applied along a trajectory over a window of W points, where half of the points are applied before the control plane and half are applied after the control plane. In the cupolet control scheme in previous chapters, an instantaneous jump was made along the control plane, but SCUTA will distribute the impact of those jumps over W samples, so that the transition is approximately smooth and continuous. In this manner, a smooth signal can be better related to biologically-accurate neural signals. As a result, the smooth signal represents the incoming pulses that a nearby neuron sends to a chaotic neuron to stabilize that chaotic neuron onto a cupolet. Figure 6.1 provides a schematic of SCUTA for illustration.

In order to apply SCUTA, the time series for a target cupolet must be known. First, using the time series of the target cupolet, locate the sampled point that is $W/2$ samples earlier than the sample directly before the control plane. This corresponds to point θ_0 in Figure 6.1. At the same time, locate the point that is $W/2$ samples after the sample directly before the control plane. This point corresponds to ϕ_6 in Figure 6.1. After these points are identified, integrate the neuron model forward W steps from θ_0 to locate θ_6 and integrate backward W steps from ϕ_6 to locate ϕ_0. Then, define $\gamma_i = \theta_i + \rho_i(\phi_i - \theta_i)$, where $\rho_i = \frac{i}{W}$ for $i = 1 \ldots W$ with $\gamma_0 = \theta_0$ and $\gamma_W = \phi_W$. For a trajectory starting on θ, specifically at θ_i, the addition of γ_i for $i = 1 \ldots W$ will result in an almost smooth interpolated transition from θ, the original trajectory, to ϕ, the target perturbed trajectory. In the follow section, the initial results of SCUTA applied to the cupolets C001 and C1010010 with a window of $W = 64$ are examined. A window of $W = 64$ corresponds to 0.5 time units since the original integration scheme has $dt = 1/128$.

6.3 Examples

6.3.1 SCUTA: C001

In this section, we provide the results of SCUTA applied with a window of $W = 64$ (half a time unit, $t = 0.5$) for C001. Figure 6.2 plots both the original C001 cupolet formed through

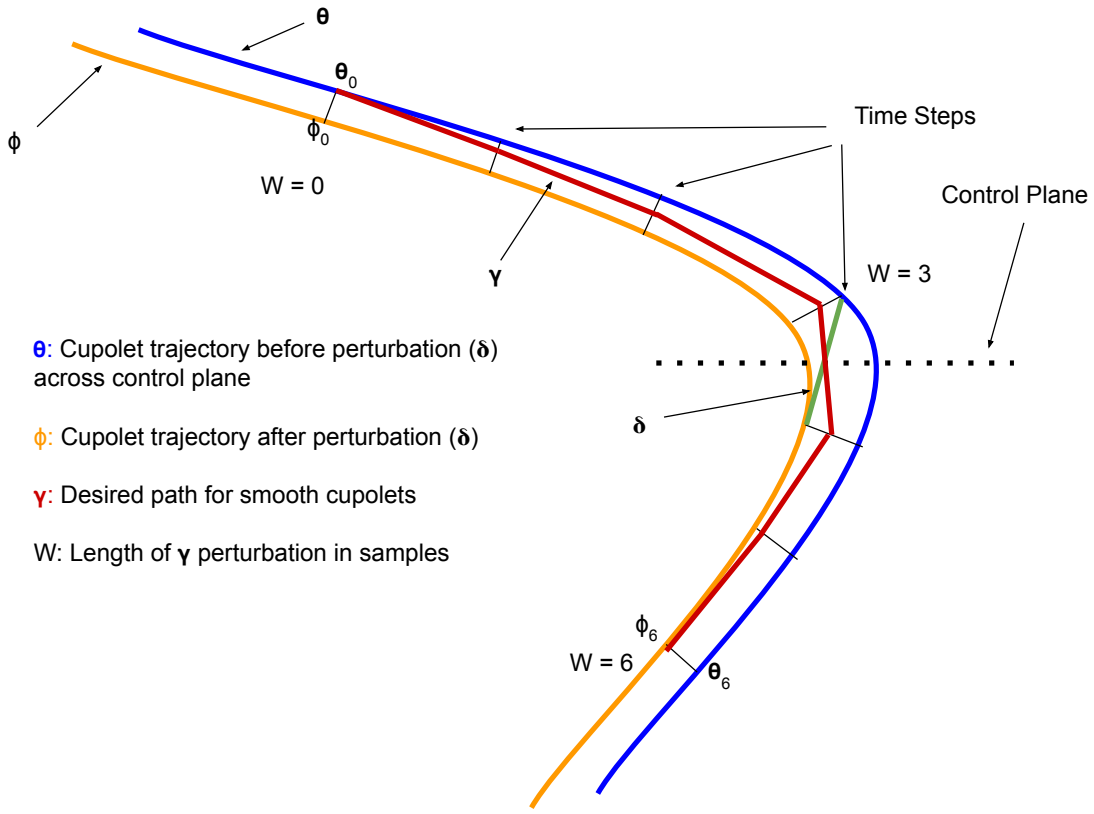

Figure 6.1: Schematic of the SCUTA algorithm applied over a window of length $W = 6$. The blue curve, labeled θ, is the trajectory of a Hindmarsh-Rose neuron without a control. The green curve, labeled δ, is the instantaneous perturbation of the blue curve such that it continues onto target trajectory represented by the orange curve, ϕ. The dashed line is the control plane where the instantaneous kicks are applied in the control scheme that generates cupolets. The thin black lines, labeled Time Steps, are the sampled points of each respective curve. The red curve, labeled γ, is the smoother interpolated trajectory after applying SCUTA.

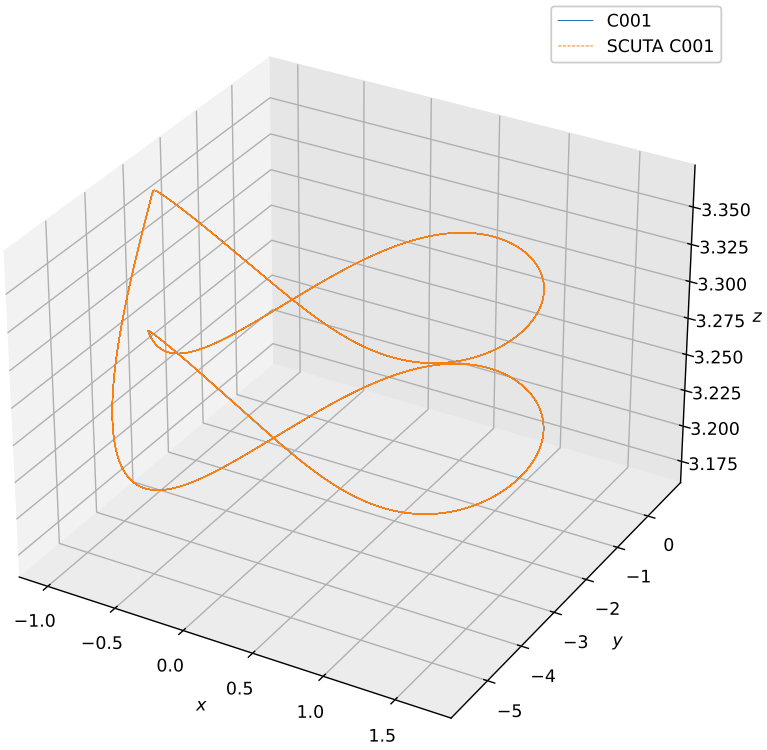

Figure 6.2: Phase plot in 3-dimensions for cupolet C001 of the Hindmarsh-Rose neuron. Blue is the cupolet formed by instantaneous perturbations through the control scheme in the previous chapters. Orange is the cupolet formed by SCUTA with a window of $W = 64$. The difference between the two is not visible at the scale of the plot.

the instantaneous perturbations (blue) and the cupolet C001 formed via SCUTA (orange). There is no visible difference between the two plots at the scale of the plot which indicates the accuracy of SCUTA.

Figure 6.3 plots several spikes from the x time series of both the original C001 (blue) and the C001 formed by SCUTA (dashed orange). Figure 6.4 gives the x values of the γ trajectory from Figure 6.1 for the first three times that SCUTA is applied across a control plane. The size of γ is small compared to the amplitude of the x values of C001 but is still large enough to smoothly push the trajectory onto the desired cupolet path.

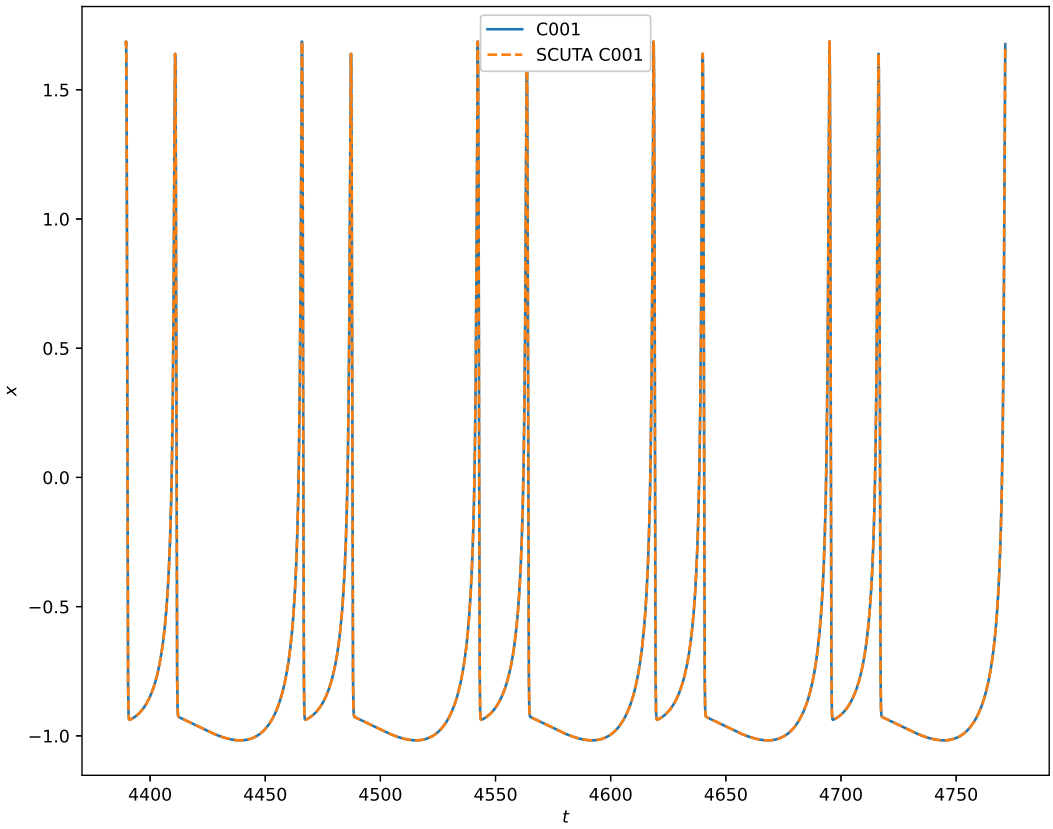

Figure 6.3: Time series of the x coordinate. Blue is cupolet C001 formed by instantaneous perturbations. The dashed orange plot is cupolet C001 formed through applying SCUTA over a window of $W = 64$.

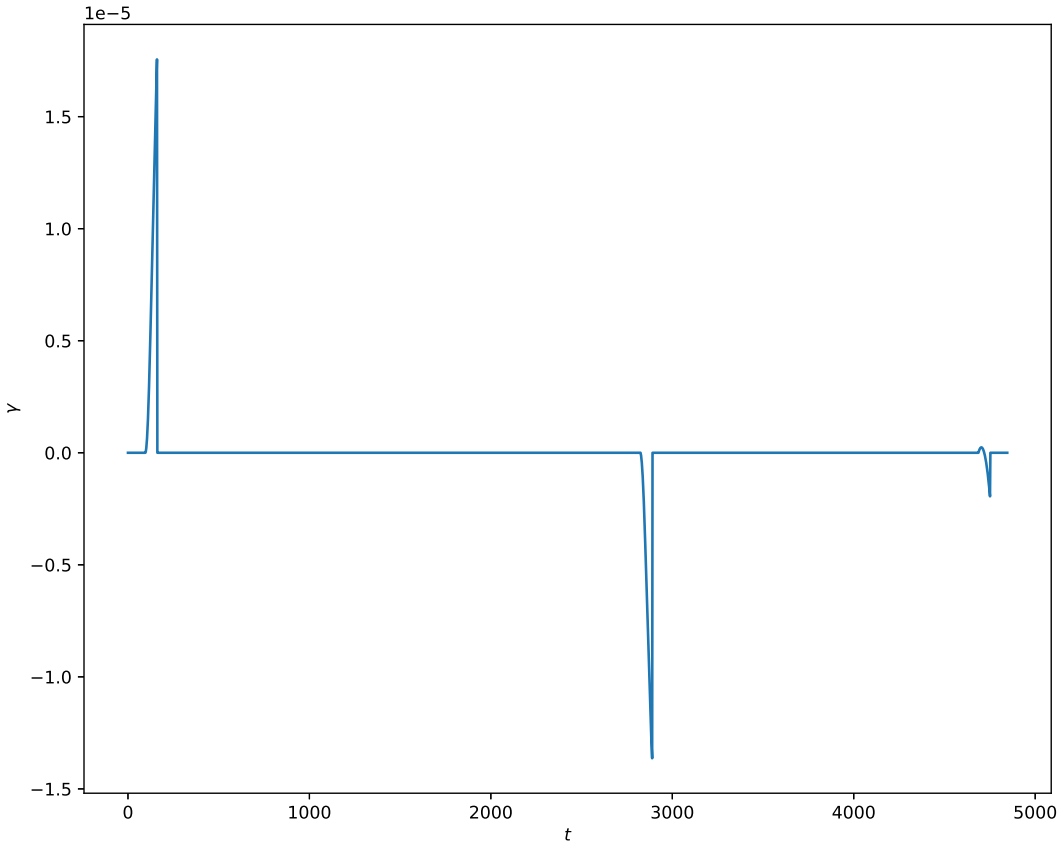

Figure 6.4: The x values of the γ time series for the first three instances that SCUTA is applied to form C001. Each pulse is the smooth transformation of the instantaneous perturbation that is applied across a control plane.

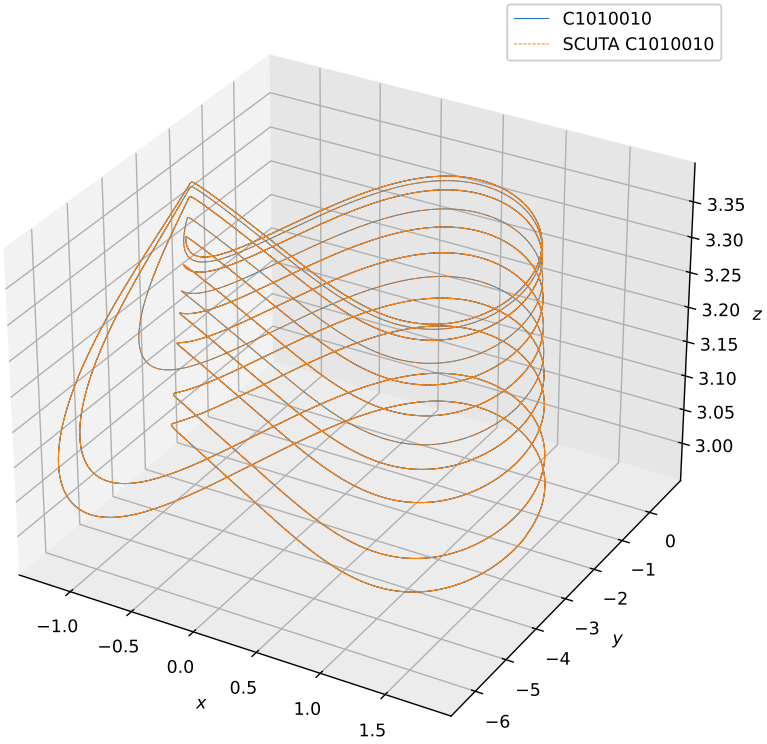

Figure 6.5: Cupolet C1010010 of the Hindmarsh-Rose neuron. Blue is the cupolet formed by instantaneous perturbations through the control scheme in the previous chapters. Orange is the cupolet formed by SCUTA with a window of $W = 64$. The difference between the two is not visible at the scale of the plot.

6.3.2 SCUTA: C1010010

In this section we provide the results of SCUTA applied with a window of $W = 64$ (half a time unit, $t = 0.5$) for cupolet C1010010. Figure 6.2 plots both the original cupolet C1010010 formed through the instantaneous perturbations (blue) and cupolet C1010010 formed via SCUTA (orange). There is no visible difference between the two plots at the scale of the plot which indicates the accuracy of SCUTA.

Figure 6.6 plots several spikes from the x variable of both the original cupolet C1010010

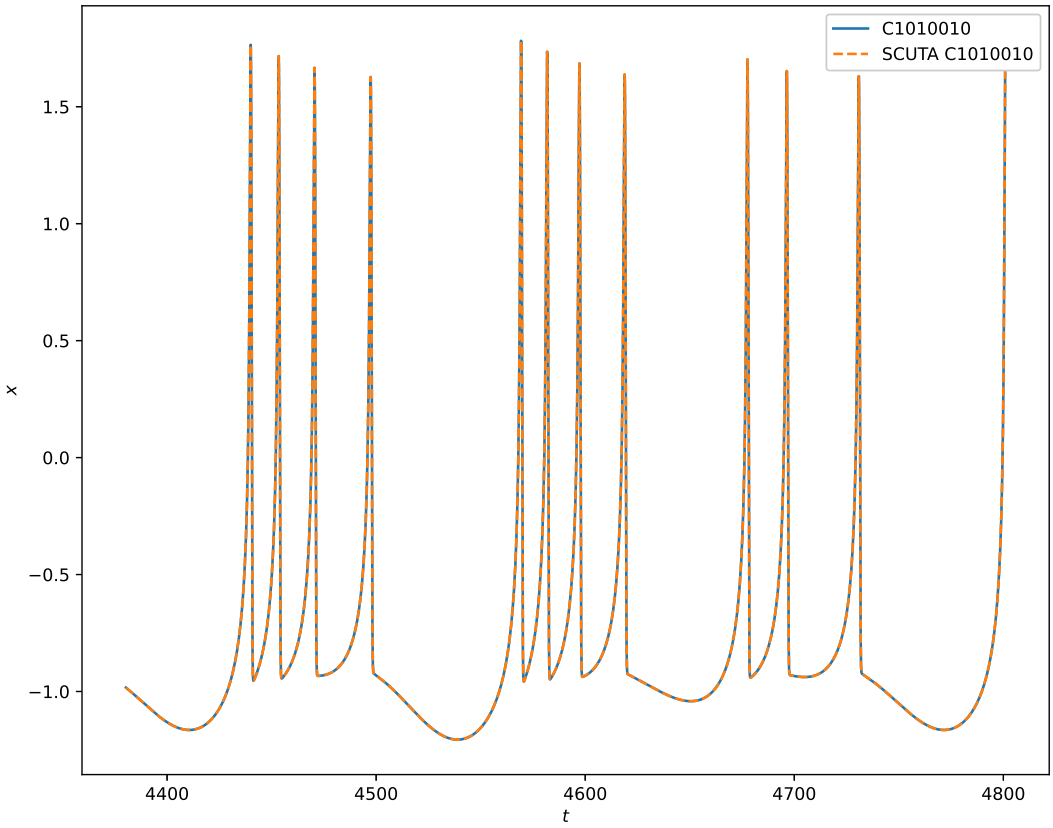

Figure 6.6: Time series of the x coordinate. Blue is cupolet C1010010 formed by instantaneous perturbations. The dashed orange plot is cupolet C1010010 formed through applying SCUTA with a window of $W = 64$.

(blue) and cupolet C1010010 formed by SCUTA (dashed orange) with a window of $W = 64$. Figure 6.7 gives the x values of the γ trajectory from Figure 6.1 for the first three times that SCUTA is applied across a control plane. The size of γ is small compared to the amplitude of the x values of C1010010 but large enough to push the trajectory onto the desired cupolet path. Note that the scale of Figure 6.7 is smaller than Figure 6.4.

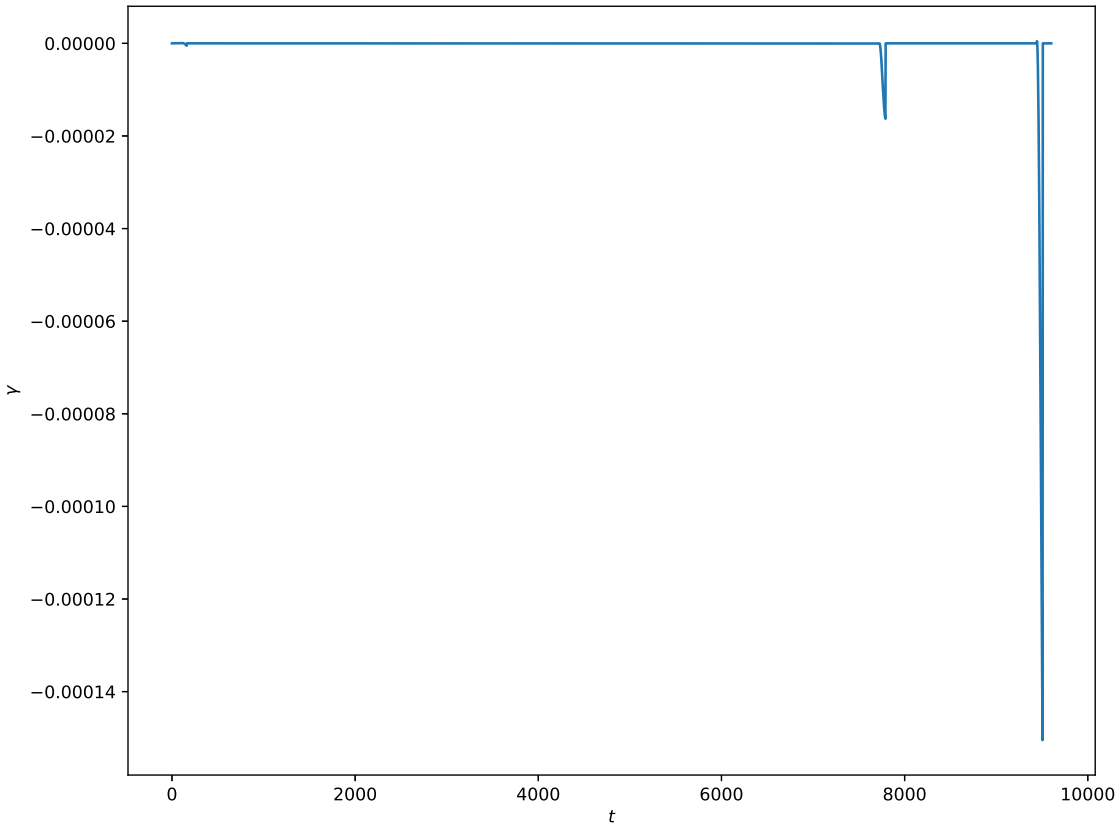

Figure 6.7: The x values of the γ time series for the first three instances that SCUTA is applied to form cupolet C1010010. Each pulse is the smooth transformation of the instantaneous perturbation that is applied across a control plane. Note that the first pulse is tiny and almost invisible.

6.4 SCUTA Challenges and Remaining Issues

In the previous section, we demonstrated the effectiveness of SCUTA with a fixed W. However, drawbacks exist for the early stages of this algorithm. Recall that the goal of SCUTA is to turn the instantaneous perturbations into smooth signals which reflect a biological signal. Therefore, the shape of the smooth signal that SCUTA generates, γ, is crucial. Currently, the shape of γ for each corresponding cupolet is dependent on two parameters, the window size W and ρ. The window size W is the number of iterations over which SCUTA is applied. The parameter ρ is the proportion of the vector difference $\phi_i - \theta_i$ that is covered at each iteration of SCUTA, and that proportion is increased with each of the time steps in Figure 6.1 before ultimately ending on the target cupolet trajectory. Currently, ρ is defined as a linearly increasing proportion ($\rho_i = i/W$ for $i = 1 \dots W$), but future work will explore making this transition more smoothly.

Figure 6.8 demonstrates the importance of the window size W on the shape of γ. Figure 6.8A shows the x values of the γ time series for cupolet C001 generated with SCUTA over a window of $W = 64$ while Figure 6.8B sets $W = 128$. Then, Figure 6.8C changes the window to $W = 256$ while Figure 6.8D sets $W = 512$. In each of these subfigures, ρ is consistent with the corresponding window size W. It can be clearly seen that the size of W impacts the shape of γ. For example, in Figure 6.8A each pulse is primarily either positive or negative. However, in Figure 6.8B and Figure 6.8C show pulses with significant positive and negative components. This is especially the case in Figure 6.8C where each pulse has a largely visible positive amount and a largely visible negative amount. This trend would continue for increasing W up to a certain limit. This limit can be seen in Figure 6.8D, where one pulse is much larger in amplitude than the other two pulses. Additionally, the pulse in Figure 6.8D is much larger in amplitude than all of the pulses in the other subfigures in Figure 6.8. The pulse in Figure 6.8D is much larger because it is applied over PS0 with a very large window that corresponds to 4 time units. During this pulse, the SCUTA C001

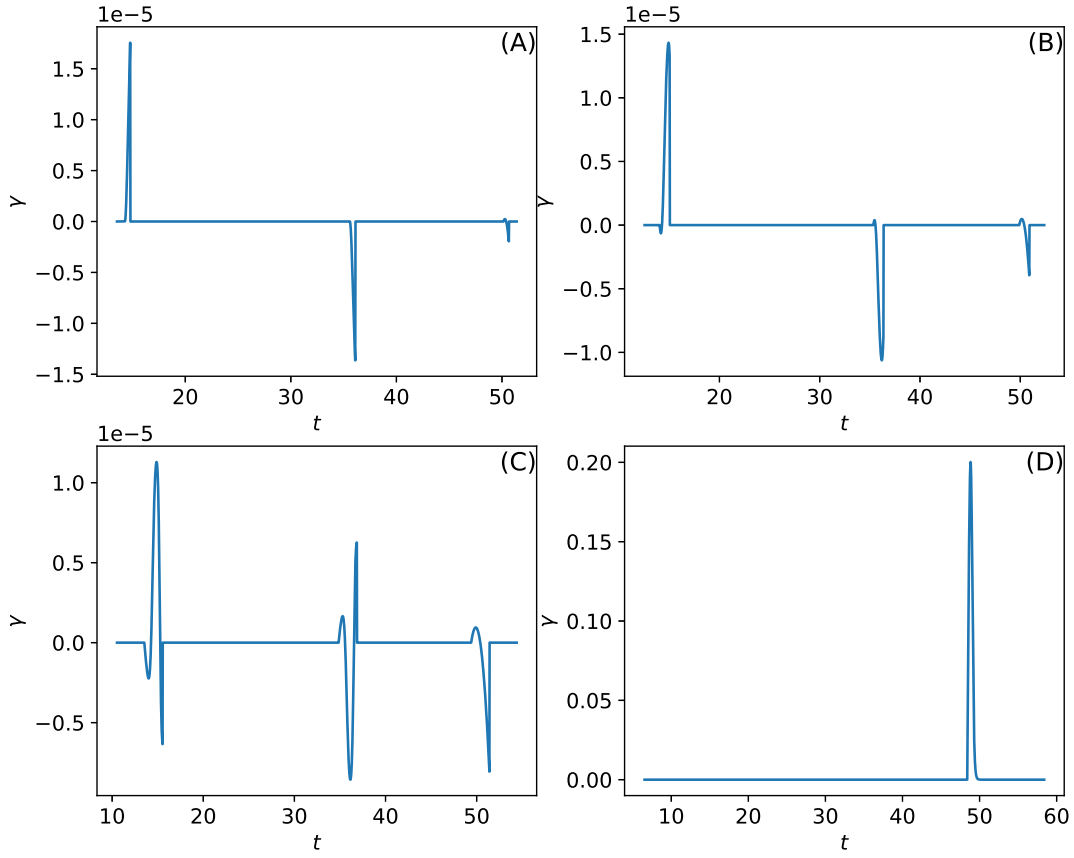

Figure 6.8: The x values of the γ time series for the first three impulses that SCUTA applies to form cupolet C001 for different window W lengths. (A) $W = 64$, (B) $W = 128$, (C) $W = 256$, (D) $W = 512$.

difference is large enough that a significant departure from the desired cupolet C001 trajectory is visible, as shown in Figure 6.9. This indicates that certain window sizes can be too large and do not guarantee a small difference with the targeted cupolet. Therefore, this implies that with the current implementation of SCUTA, only certain W sizes are permitted for appropriate smooth pulses.

The parameter ρ can be given a functional form to further augment the pulses that SCUTA implements. This parameter can be considered as a distribution that sums to 1. If the distribution were properly adjusted then different pulse shapes would be formed since

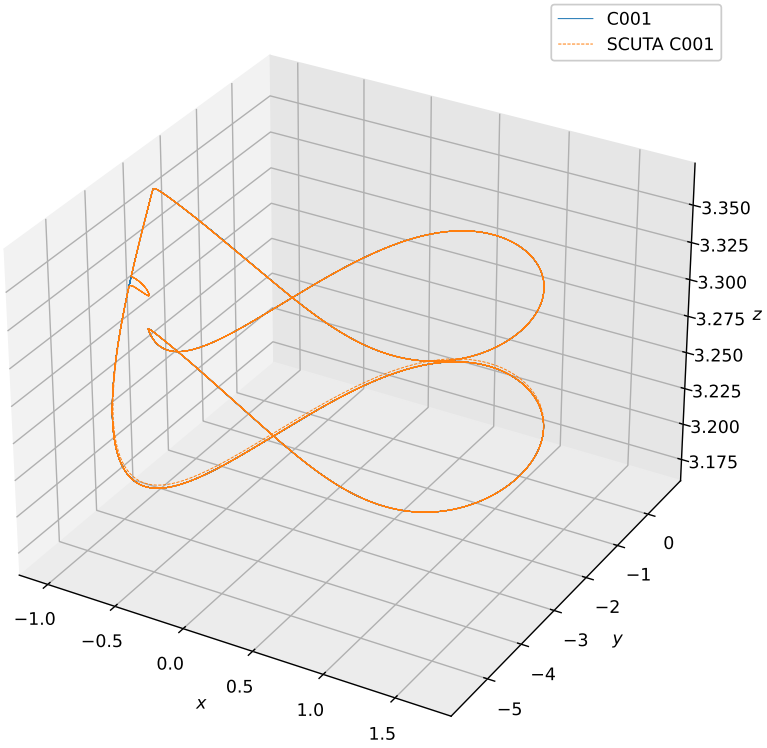

Figure 6.9: Phase plot in 3-dimensions for cupolet C001 of the Hindmarsh-Rose neuron. Blue is the cupolet formed by instantaneous perturbations through the control scheme in the previous chapters. Orange is the cupolet formed by SCUTA with a window of $W = 512$ that leads to a distinct difference between the two plots, particularly near control plane PS0. This difference indicates that there is an upper limit for a possible window size, W.

certain iterations could correspond to larger deviations. For example, consider $W = 64$, but only have $\rho_{32} = 1$ while all other ρ_i values are 0. The entire length of the pulse is applied over 1 step, similar to the instantaneous perturbations that originally generated the Hindmarsh-Rose cupolets discussed in this thesis. However, if we were to choose a proportion that is chosen equally across a given distribution, then normalized to one, so that the largest jumps would be in the middle between trajectories, then we would expect smooth Gaussian like pulses. There are many possible distributions for ρ but both W and ρ must be properly adjusted for correct implementation of smooth biological-accurate signals that generate cupolets.

6.5 Conclusion

We have reported on an initial algorithm, called Smooth Cupolet Targeting Algorithm (SCUTA), that allows for small pulses to be applied to a Hindmarsh-Rose trajectory over a given width window of W samples. The small pulses represent a more biologically-accurate incoming signal from a nearby neuron that stabilizes a chaotic neuron onto a cupolet. If the trajectory is lined up correctly then SCUTA will allow for a smooth transition towards a targeted cupolet. The examples of cupolet C001 and cupolet C1010010 are given where SCUTA is applied over a window of $W = 64$ integration steps corresponding to an integration time of 0.5 time units. In each example, the pulses are small relative to the size of the trajectory, yet still large enough to shift the trajectory onto the target cupolet. SCUTA allows for continuous pulses to be constructed that elicit the same results as Chapter 3, assuming the points θ_0 and ϕ_W are already known previous to generation of the desired cupolet. Thus, SCUTA provides a foundation to extend the discrete binary controls and interaction functions from the previous chapters into continuous signals.

We addressed drawbacks of SCUTA concerning W and ρ in §6.4. We suggest that a machine learning approach may best determine what the proper W and ρ parameters that can most accurately reproduce a given biological signal. Additionally, another drawback of

SCUTA is that the cupolet target and future trajectory of the cupolet needs to be known, although this is not an impediment if the equations are available. In this initial implementation, SCUTA is very sensitive to the phase of the pulses relative to the control planes, so if SCUTA is not applied exactly at ϕ_0 and θ_0 then the trajectory will not be pushed onto the desired target cupolet as accurately and the system may drift back into chaos. This is expected due to the sensitivity of initial conditions that is typical of chaotic behavior, which is the default behavior for the parameter set we are using for the HR neural system. Future work will focus on generalizing the SCUTA approach so that a periodic sequence of pulses can be sent into the input of a chaotic HR neuron to drive the HR neuron into a persistent, periodic trajectory of a cupolet in a manner where the only controls are those coming from the pulses transmitted from nearby neurons. Additionally, we will explore ways to exploit the dynamics of the HR system such that SCUTA can be applied without previous knowledge of the target cupolet.

CHAPTER 7
CONCLUDING REMARKS

The focus of this thesis has been on mathematical models of neurons that are capable of periodic and chaotic behavior, where the neural firing was either single spiking (FHN model) or bursts of spikes (HR model). The major result of this thesis is that chaotic neurons are capable of stabilizing onto periodic orbits, called cupolets, and communicating neurons can share information about stabilized periodic orbits such that mutual stabilization can occur. The stabilization onto periodic orbits can occur through binary control sequences, interactions with other neurons, or smooth pulses. The incoming signal, whether binary or continuous, is representative of the input that a neuron receives. The input alters the membrane potential of the neuron by changing the neural dynamics of the neuron.

In Chapter 1 and Chapter 2, it was shown how sigmoidal synaptic learning might lead a bidirectional FitzHugh-Nagumo neural system away from a chaotic regime and into a region of parameter space where mutual stabilization can occur. Then, the results from Chapter 3 to Chapter 5 examined how interactions between neurons, specifically chaotic Hindmarsh-Rose neurons, can lead to forms of stabilization, even though the neurons exist in a chaotic regime of parameter space. The four different versions of stabilization discussed in this thesis were cupolets, unidirectional stabilization, mutual stabilization, chain mutual stabilization, and feedback stabilization. The last chapter, Chapter 6, reported on the findings of an initial algorithm, SCUTA, that transforms the instantaneous pulses that led to chaotic stabilization onto periodic orbits into smoother, more biologically-accurate pulses that retain the capability to stabilize a chaotic neuron onto a cupolet.

The stabilization of a chaotic neuron to periodic firing has several implications. The communicating neurons retain information about an incoming signal and this information allows for persistent, periodic dynamics. This suggests that a form of "stored memory" might be occurring during forms of neural stabilization. Additionally, this thesis has set the foundation for an investigation into larger networks of neurons to determine whether any further properties might occur when similar methods of stabilization are applied. Other future work will seek to examine how a system evolving with persistent, periodic behavior can extract the retained information as well as examine how synchronization and network behaviors can be controlled in large networks.

Bibliography

1. Hunt, B. R. & Ott, E. Defining chaos. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **25,** 097618. arXiv: 1501.07896 (Sept. 2015).
2. Poincaré, H. Sur le problème des trois corps et les équations de la dynamique. *Acta Mathematica* **13,** 1–271 (Dec. 1890).
3. Parker, A. T. *Topics in Chaotic Secure Communication* PhD thesis (University of New Hampshire, 1999).
4. Morena, M. & Short, K. Cupolets and a Chaotic Analog of Entanglement. *arXiv:1302.2283,* 1–12. arXiv: 1302.2283 (Feb. 2013).
5. Zarringhalam, K. *CUPOLETS: Chaotic unstable periodic orbits theory and applications* PhD thesis (University of New Hampshire, 2006).
6. Zarringhalam, K. & Short, K. M. Generating an adaptive multiresolution image analysis with compact cupolets. *Nonlinear Dynamics* **52,** 51–70 (Apr. 2008).
7. Morena, M. A. & Short, K. M. On the Potential for Entangled States Between Chaotic Systems. *International Journal of Bifurcation and Chaos* **24,** 1450077 (June 2014).
8. Short, K. M. & Morena, M. A. Signatures of Quantum Mechanics in Chaotic Systems. *Entropy* **21,** 618 (June 2019).
9. Hayes, S., Grebogi, C. & Ott, E. Communicating with chaos. *Physical Review Letters* **70,** 3031–3034 (May 1993).
10. Hayes, S., Grebogi, C., Ott, E. & Mark, A. Experimental Control of Chaos for Communication. *Physical Review Letters* **73,** 1781–1784 (Sept. 1994).
11. Hodgkin, A. L. & Huxley, A. F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology* **117,** 500–544 (Aug. 1952).
12. Fan, Y.-S. & Chay, T. R. Generation of periodic and chaotic bursting in an excitable cell model. *Biological Cybernetics* **71,** 417–431 (Sept. 1994).
13. FitzHugh, R. Impulses and Physiological States in Theoretical Models of Nerve Membrane. *Biophysical Journal* **1,** 445–466 (July 1961).
14. Nagumo, J., Arimoto, S. & Yoshizawa, S. An Active Pulse Transmission Line Simulating Nerve Axon. *Proceedings of the IRE* **50,** 2061–2070 (Oct. 1962).
15. Vaidyanathan, S. Global chaos control of the FitzHugh-Nagumo chaotic neuron model via integral sliding mode control. *International Journal of PharmTech Research* **9,** 413–425 (2016).

16. Lai, T.-W., Lin, J.-S., Liao, T.-L. & Yan, J.-J. in *Communications in Computer and Information Science* (eds Park, J.-W., Kim, T.-G. & Kim, Y.-B.) 142–150 (Springer Berlin Heidelberg, 2007).

17. Campbell, S. A. & Waite, M. Multistability in coupled Fitzhugh-Nagumo oscillators. *Nonlinear Analysis: Theory, Methods & Applications* **47,** 1093–1104 (Aug. 2001).

18. Parker, J. E. & Short, K. M. Sigmoidal synaptic learning produces mutual stabilization in chaotic FitzHugh-Nagumo model. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **30,** 063108 (June 2020).

19. Brown, D., Feng, J. & Feerick, S. Variability of Firing of Hodgkin-Huxley and FitzHugh-Nagumo Neurons with Stochastic Synaptic Input. *Physical Review Letters* **82,** 4731–4734 (June 1999).

20. Rehan, M. & Hong, K. S. LMI-based robust adaptive synchronization of FitzHugh-Nagumo neurons with unknown parameters under uncertain external electrical stimulation. *Physics Letters, Section A: General, Atomic and Solid State Physics* **375,** 1666–1670 (Apr. 2011).

21. Skinner, F. K., Zhang, L., Velazquez, J. L. P. & Carlen, P. L. Bursting in Inhibitory Interneuronal Networks: A Role for Gap-Junctional Coupling. *Journal of Neurophysiology* **81,** 1274–1283 (Mar. 1999).

22. Chillemi, S., Barbi, M. & Di Garbo, A. The role of synaptic coupling in a network of FHN neuron models. *Nonlinear Analysis: Theory, Methods & Applications* **47,** 2163–2169 (Aug. 2001).

23. Hoff, A., dos Santos, J. V., Manchein, C. & Albuquerque, H. A. Numerical bifurcation analysis of two coupled FitzHugh-Nagumo oscillators. *The European Physical Journal B* **87,** 151. arXiv: `arXiv:1504.07565v1` (July 2014).

24. Davison, E. N., Aminzare, Z., Dey, B. & Ehrich Leonard, N. Mixed mode oscillations and phase locking in coupled FitzHugh-Nagumo model neurons. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **29,** 033105. arXiv: `1807.10824` (Mar. 2019).

25. Ao, X., Hänggi, P. & Schmid, G. In-phase and anti-phase synchronization in noisy Hodgkin-Huxley neurons. *Mathematical Biosciences* **245,** 49–55 (Sept. 2013).

26. Shuai, J.-W. & Durand, D. M. Phase synchronization in two coupled chaotic neurons. *Physics Letters A* **264,** 289–297 (Dec. 1999).

27. Hebb, D. O. *The Organization of Behavior: A Neuropsychological Theory* (Wiley, 1949).

28. Oja, E. Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology* **15,** 267–273 (Nov. 1982).

29. Virtanen, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods* **17,** 261–272 (Mar. 2020).

30. Oliphant, T. E. Python for Scientific Computing. *Computing in Science & Engineering* **9,** 10–20 (2007).

31. Van der Walt, S., Colbert, S. C. & Varoquaux, G. The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science & Engineering* **13,** 22–30 (Mar. 2011).

32. Petzold, L. Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. *SIAM Journal on Scientific and Statistical Computing* **4,** 136–148 (1983).

33. Hindmarsh, A. C. ODEPACK, A Systematized Collection of ODE Solvers. *IMACS Transactions on Scientific Computation* **1,** 55–64 (1983).

34. Deleuze, C. *et al.* Strong preference for autaptic self-connectivity of neocortical PV interneurons facilitates their tuning to γ-oscillations. *PLOS Biology* **17** (ed Csicsvari, J.) e3000419 (Sept. 2019).

35. Li, Y., Schmid, G., Hänggi, P. & Schimansky-Geier, L. Spontaneous spiking in an autaptic Hodgkin-Huxley setup. *Physical Review E* **82,** 061907 (Dec. 2010).

36. Tessone, C. J., Ullner, E., Zaikin, A. A., Kurths, J. & Toral, R. Noise-induced inhibitory suppression of frequency-selective stochastic resonance. *Physical Review E* **74,** 046220 (Oct. 2006).

37. Morris, C. & Lecar, H. Voltage oscillations in the barnacle giant muscle fiber. *Biophysical Journal* **35,** 193–213 (July 1981).

38. Chay, T. R. Chaos in a three-variable model of an excitable cell. *Physica D: Nonlinear Phenomena* **16,** 233–242 (June 1985).

39. Hindmarsh, J. L. & Rose, R. M. A Model of Neuronal Bursting Using Three Coupled First Order Differential Equations. *Proceedings of the Royal Society of London. Series B. Biological Sciences* **221,** 87–102 (Mar. 1984).

40. Izhikevich, E. M. Which Model to Use for Cortical Spiking Neurons? *IEEE Transactions on Neural Networks* **15,** 1063–1070 (Sept. 2004).

41. Guckenheimer, J. & Oliva, R. A. Chaos in the Hodgkin–Huxley Model. *SIAM Journal on Applied Dynamical Systems* **1,** 105–114 (Jan. 2002).

42. Rabinovich, M. & Abarbanel, H. The role of chaos in neural systems. *Neuroscience* **87,** 5–14 (June 1998).

43. Faure, P. & Korn, H. Is there chaos in the brain? I. Concepts of nonlinear dynamics and methods of investigation. *Comptes Rendus de l'Académie des Sciences - Series III - Sciences de la Vie* **324,** 773–793 (Sept. 2001).

44. Korn, H. & Faure, P. Is there chaos in the brain? II. Experimental evidence and related models. *Comptes Rendus Biologies* **326,** 787–840 (Sept. 2003).

45. Erichsen, R., Mainieri, M. S. & Brunnet, L. G. Periodicity and chaos in electrically coupled Hindmarsh-Rose neurons. *Physical Review E* **74,** 061906 (Dec. 2006).

46. Doungmo Goufo, E. F. & Tabi, C. B. On the chaotic pole of attraction for Hindmarsh-Rose neuron dynamics with external current input. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **29,** 023104 (Feb. 2019).

47. Cvitanović, P. Periodic orbits as the skeleton of classical and quantum chaos. *Physica D: Nonlinear Phenomena* **51,** 138–151 (Aug. 1991).

48. Morena, M. A. *Mutual stabilization of chaotic systems through entangled cupolets* PhD thesis (University of New Hampshire, 2014).

49. Morena, M. A., Short, K. M. & Cooke, E. E. Controlled transitions between cupolets of chaotic systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **24,** 013111 (Mar. 2014).

50. Morena, M. A. & Short, K. M. Fundamental cupolets of chaotic systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **30,** 093114 (Sept. 2020).

51. Matsumoto, T., Chua, L. O. & Komuro, M. The double scroll. *IEEE Transactions on Circuits and Systems* **32,** 797–818 (Aug. 1985).

52. Short, K. M., Garcia, R. A., Daniels, M., Curley, J. & Glover, M. *An introduction to the KOZ scalable audio compression technology* in *Audio Engineering Society - 118th Convention Spring Preprints 2005* **1** (Barcelona, Spain, 2005), 432–442.

53. Short, K. M., Garcia, R. A. & Daniels, M. L. *Scalability in KOZ audio compression technology* in *Audio Engineering Society - 119th Convention Fall Preprints 2005* **1** (New York, NY, 2005), 176–185.

54. Innocenti, G., Morelli, A., Genesio, R. & Torcini, A. Dynamical phases of the Hindmarsh-Rose neuronal model: Studies of the transition from bursting to spiking chaos. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **17,** 043128 (Dec. 2007).

55. Innocenti, G. & Genesio, R. On the dynamics of chaotic spiking-bursting transition in the Hindmarsh–Rose neuron. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **19,** 023124 (June 2009).

56. Storace, M., Linaro, D. & de Lange, E. The Hindmarsh–Rose neuron model: Bifurcation analysis and piecewise-linear approximations. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **18,** 033128 (Sept. 2008).

57. Henon, M. On the numerical computation of Poincaré maps. *Physica D: Nonlinear Phenomena* **5,** 412–414 (Sept. 1982).

58. Lapicque, L. Recherches quantitatives sur l'excitation electrique des nerfs traitee comme une polarization. *Journal of Physiol Pathol Générale* **9,** 620–635 (Dec. 1907).

59. Brunel, N. & van Rossum, M. C. W. Quantitative investigations of electrical nerve excitation treated as polarization. *Biological Cybernetics* **97,** 341–349 (Dec. 2007).

60. Racicot, D. M. & Longtin, A. Interspike interval attractors from chaotically driven neuron models. *Physica D: Nonlinear Phenomena* **104,** 184–204 (May 1997).

61. Hall, G. M., Bahar, S. & Gauthier, D. J. Experimental control of a chaotic point process using interspike intervals. *Physical Review E* **58,** 1685–1689 (Aug. 1998).